МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проєктами
Кафедра програмного забезпечення автоматизованих систем
Спеціальність 124 «Системний аналіз»
Освітня програма «Інформаційні технології інтелектуального аналізу даних та проєктування програмних систем»

«Допущений до захисту»

Завідувач кафедри

_____ проф. Приходько С.Б.

«___» _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття ступеня вищої освіти «магістр»

на тему: Математична модель раннього прогнозування розміру програмних систем під управлінням платформи WordPress та програма для її реалізації

Виконав: студент 6153м групи
Проценко В.С._____
(підпис, ПІБ)

Керівник роботи:
доц. каф. ПЗАС, к.т.н., доц.___
(посада, науковий ступень вчене звання)
Каіров В.О. _____
(підпис, ПІБ)

Миколаїв - 2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проєктами
Кафедра програмного забезпечення автоматизованих систем
Спеціальність 124 «Системний аналіз»
Освітня програма «Інформаційні технології інтелектуального аналізу даних та проєктування програмних систем»

«ЗАТВЕРДЖУЮ»
Гарант освітньої програми
_____доц. Латанська Л.О.
(підпис)
«15» 10 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти «магістр»

Студенту Проценко Віталію Сергійовичу
(Прізвище, ім'я, по батькові)

1. Тема роботи: Математична модель раннього прогнозування розміру програмних систем під управлінням платформи WordPress та програма для її реалізації
Керівник роботи Каіров В.О, доц. каф. ПЗАС, к.т.н., доц.
Затверджені наказом ректора № 1222-уч від «15» 10 2025 р.
2. Термін подання роботи: 08.12.2025 р.
3. Вихідні дані по роботі: _____
4. Перелік питань, що належать до розробки (найменування розділів)

Вступ (Актуальність теми. Зв'язок роботи з науковими програмами, планами, темами. Мета і завдання дослідження. Об'єкт дослідження. Предмет дослідження. Методи дослідження. Наукова новизна одержаних результатів. Практичне значення одержаних результатів. Особистий внесок здобувача. Апробація результатів досліджень**.** Публікації.); Огляд літератури та обґрунтування необхідності проведення досліджень за обраною темою; Викладення результатів власних досліджень з висвітленням того нового, що пропонується; Проєкт програмного забезпечення; Організаційно-економічний розділ; Розділи з охорони праці та охорони навколишнього середовища; Висновки; Список використаних джерел; Додатки (технічне завдання, текст програми, опис програми, інструкція користувача, програма і методика випробувань програмного забезпечення)

5. Перелік презентаційних матеріалів: Титульний слайд, Мета і завдання дослідження, Актуальність теми, Об'єкт і предмет дослідження, Методи дослідження, Наукова новизна одержаних результатів, Практичне значення одержаних результатів, Апробація результатів досліджень та публікації, Нелінійна регресійна модель, Постановка задачі, Загальносистемні рішення, Інформаційне забезпечення, Програмне забезпечення, Висновки

6. Консультанти розділів роботи

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

7. Дата видачі завдання 15.10.2025 р.

К А Л Е Н Д А Р Н И Й П Л А Н

Номер	Назва етапів роботи	Терміни виконання	Примітка
1.	Підготовка розділу Вступ	28.10.2025	ДП*
2.	Підготовка розділу з огляду літератури та обґрунтування необхідності проведення досліджень за обраною темою	30.10.2025	ДП*
3.	Підготовка розділу (ів) з результатів власних досліджень	01.11.2025	ДП*
4.	Підготовка розділу з Проєкту програмного забезпечення	26.11.2025	
5.	Підготовка організаційно-економічного розділу	28.11.2025	
6.	Підготовка розділу з охорони праці	02.12.2025	
7.	Підготовка розділу з охорони навколишнього середовища	03.12.2025	
8.	Підготовка розділу Висновки	04.12.2025	
9.	Оформлення списку використаних джерел та додатків	05.12.2025	
10.	Подання на кафедру ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, у роздрукованому та електронному форматі разом із заявами щодо самостійності виконання роботи та ідентичності друкованої та електронної версії роботи (Додатки 1 і 2 «Порядку здійснення заходів з перевірки робіт на наявність текстових збігів/ідентичності/схожості із використанням програмно-технічних засобів», який введений в дію наказом ректора НУК за №20 від 20.01.2020 р.)	08.12.2025	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку
11.	Підготовка презентації та доповіді	12.12.2025	
12.	Попередній захист роботи на засіданні кафедри ПЗАС	15.12.2025	
13.	Подання на кафедру ПЗАС електронних версії наступних документів у форматі pdf: кваліфікаційної роботи; файлу-опису кваліфікаційної роботи (згідно Додатку до наказу ректора НУК за №287-уч від 19.05.2020 р.); презентації доповіді	19.12.2025	

* - за результатами (розділ звіту) дослідницької практики (ДП), яка була з 01.09.2025 по 26.10.2025 р.

Студент _____ Проценко В.С._____
(підпис) (ПІБ)

Керівник роботи _____ Каіров В.О._____
(підпис) (ПІБ)

РЕФЕРАТ

Проценко Віталій Сергійович

Математична модель раннього прогнозування розміру програмних систем під управлінням платформи WordPress та програма для її реалізації

Кваліфікаційна робота на здобуття освітнього рівня магістра зі спеціальності 124 - «Системний аналіз». Національний університет кораблебудування імені адмірала Макарова. Миколаїв, 2025 р.

Обсяг роботи: 83 стор., 9 табл., 9 рис., 19 використаних джерел, 5 додатків.

Актуальність теми роботи: полягає в тому, що раннє прогнозування розміру майбутнього проєкту є актуальною задачею, виходячи з поширеності цієї системи керування вмістом.

Мета та завдання дослідження. Метою роботи є підвищення достовірності раннього прогнозування розміру програмних систем, створених під управлінням платформи WordPress мовою програмування PHP. Завдання: проаналізувати існуючі моделі раннього прогнозування розміру програмних систем, зокрема під платформу WordPress; обґрунтувати необхідність побудови математичної моделі для раннього прогнозування розміру програмних систем для WordPress; побудувати нелінійну регресійну модель прогнозування розміру програмних систем для WordPress із використанням нормалізуючого перетворення; розробити програму, що реалізує побудовану модель для практичного застосування при оцінюванні розміру програмних систем.

Об'єктом дослідження є процес раннього прогнозування розміру програмних систем, створених під управлінням платформи WordPress.

Предметом дослідження є нелінійна регресійна модель раннього прогнозування розміру програмних систем створених під управлінням платформи WordPress.

Методи дослідження: математичного моделювання, теорії ймовірності, математичної статистики та регресійного аналізу, об`єктно-орієнтованого програмування.

Наукова новизна одержаних результатів полягає в удосконаленні математичної моделі раннього прогнозування розміру програмних систем під управлінням платформи Wordpress шляхом побудови нелінійної регресійної моделі з нормалізуючим перетворенням у вигляді десяткового логарифму. Це забезпечує підвищення достовірності раннього прогнозування розміру у порівнянні з існуючими моделями.

Практичне значення одержаних результатів. Розроблено програму, яка реалізує побудовану математичну модель раннього прогнозування розміру програмних систем під управлінням платформи WordPress.

Ключові слова: прогнозування розміру програмних систем, математична модель, рівняння регресії, нормалізуюче перетворення, WordPress, PHP, програмне забезпечення.

ABSTRACT

Protsenko Vitalii Serhiiovych

Mathematical model for early prediction of the size of software systems under the management of the Wordpress platform and a software for its implementation

Qualification work for obtaining a master's degree in specialty 124 - ‘System Analysis’. Admiral Makarov National University of Shipbuilding. Mykolaiv, 2025.

Scope of work: 83 pages, 9 tables, 9 figures, 19 sources used, 5 appendices.

Relevance of the topic: early prediction of the size of a future project is a pressing issue, given the prevalence of this content management system.

Purpose and objectives of the study. The purpose of the work is to improve the reliability of early prediction of the size of software systems created under the WordPress platform using the PHP programming language. Tasks: to analyse existing models of early prediction of the size of software systems, in particular for the WordPress platform; to justify the need to build a mathematical model for early prediction of the size of software systems for WordPress; to build a nonlinear regression model for predicting the size of software systems for WordPress using normalising transformation; to develop a program that implements the constructed model for practical application in estimating the size of software systems.

The object of the study is the process of early prediction of the size of software systems created under the WordPress platform.

The subject of the study is a nonlinear regression model for early prediction of the size of software systems created under the WordPress platform.

Research methods: mathematical modelling, probability theory, mathematical statistics and regression analysis, object-oriented programming.

The scientific novelty of the results obtained lies in the improvement of the mathematical model for early prediction of the size of software systems under the control of the WordPress platform by constructing a nonlinear regression model with a normalising transformation in the form of a decimal logarithm. This ensures an increase in the reliability of early size prediction compared to existing models.

Practical significance of the results obtained. A programme has been developed that implements the constructed mathematical model for early prediction of the size of software systems running on the WordPress platform.

Keywords: prediction of the size of software systems, mathematical model, regression equation, normalising transformation, WordPress, PHP, software

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ..... 8

ВСТУП..... 9

1 ДОСЛІДЖЕННЯ ІСНУЮЧИХ МЕТОДІВ ТА МОДЕЛЕЙ ДЛЯ РАННЬОГО ПРОНОЗУВАННЯ РОЗМІРУ ПРОГРАМНИХ СИСТЕМ ПІД УПРАВЛІННЯМ ПЛАТФОРМИ WORDPRESS..... 12

1.1 Особливості та переваги використання WordPress як платформи керування вмістом веб-сайтів..... 12

1.2 Дослідження існуючих методів та моделей для раннього прогнозування розміру програмних систем 14

1.3 Обгрунтування необхідності проведення досліджень за обраною темою..... 16

1.4 Висновки за розділом 1 17

2 РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ ДЛЯ РАННЬОГО ПРОГНОЗУВАННЯ РОЗМІРУ ПРОГРАМНИХ СИСТЕМ ПІД УПРАВЛІННЯМ ПЛАТФОРМИ WORDPRESS..... 18

2.1 Аналіз наявних математичних моделей та вибір факторів для раннього прогнозування розміру програмних систем під управлінням платформи WordPress 18

2.2 Попередня обробка емпіричних даних для раннього прогнозування розміру програмних систем під управлінням платформи WordPress 20

2.3 Побудова нелінійної регресійної моделі для раннього прогнозування розміру програмних систем під управлінням платформи WordPress 26

2.4 Висновки за розділом 2 30

3 РОЗРОБКА ПРОЄКТНИХ РІШЕНЬ ДЛЯ ПРОГРАМНОЇ СИСТЕМИ РАННЬОГО ПРОГНОЗУВАННЯ РОЗМІРУ ПС ПІД УПРАВЛІННЯМ ПЛАТФОРМИ WORDPRESS..... 31

3.1 Постановка задачі на розробку програмної системи раннього прогнозування розміру ПС під управлінням платформи WordPress..... 31

3.2 Загальносистемні рішення для програмної системи раннього прогнозування розміру ПС під управлінням платформи WordPress..... 32

3.2.1 Вибір засобів моделювання для ПС прогнозування розміру..... 32

3.2.2 Розробка діаграми варіантів використання ПС прогнозування розміру 33
3.2.3 Специфікації варіантів використання ПС прогнозування розміру 34
3.2.4 Сценарії основних варіантів використання ПС прогнозування розміру 36
3.3 Рішення з інформаційного забезпечення для програмної системи прогнозування розміру 38
3.3.1 Концептуальна модель даних ПС прогнозування розміру 38
3.3.2 Логічна модель даних ПС прогнозування розміру 39
3.4 Рішення з програмного забезпечення програмної системи прогнозування розміру 39
3.4.1 Вибір мови програмування для ПС прогнозування розміру 39
3.4.2 Модель класів ПС прогнозування розміру 40
3.4.3 Моделі взаємодії ПС прогнозування розміру..... 42
3.4.4 Тестування ПС прогнозування розміру 43
3.4.5 Випробування ПС прогнозування розміру 43
3.5 Висновки за розділом 3 44
4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД РОЗРОБКИ І ВПРОВАДЖЕННЯ ПРОГРАМНОЇ СИСТЕМИ ДЛЯ РАННЬОГО ПРОГНОЗУВАННЯ РОЗМІРУ ПС ПІД УПРАВЛІННЯМ ПЛАТФОРМИ WORDPRESS..... 46
4.1 Розрахунок витрат на створення та впровадження програмної системи 47
4.2 Розрахунок економічної ефективності розробки та впровадження програмної системи..... 50
4.3 Висновки за розділом 4 51
5 ОХОРОНА ПРАЦІ..... 52
5.1 Аналіз шкідливих та небезпечних факторів у офісі комп’ютерної фірми..... 53
5.2 Розрахунок системи пожежогасіння у офісі комп’ютерної фірми..... 57
5.3 Розробка заходів щодо зменшення впливу шкідливих та небезпечних факторів 59
6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА 61
6.1 Вступ..... 61
6.2 Аналіз існуючої проблеми необхідності охорони навколишнього середовища 62
6.3 Забруднення навколишнього середовища співробітниками офісу комп’ютерної фірми 63
6.4 Розробка заходів щодо зменшення забруднення 63
ВИСНОВКИ 65

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... 67

ДОДАТОК А - ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ ПРОГРАМНОЇ СИСТЕМИ РАННЬОГО ПРОГНОЗУВАННЯ РОЗМІРУ ПС ПІД УПРАВЛІННЯМ ПЛАТФОРМИ WORDPRESS «INFOPLUG»..... 70

ДОДАТОК Б - КОД ПРОГРАМНОЇ СИСТЕМИ «INFOPLUG» РАННЬОГО ПРОГНОЗУВАННЯ РОЗМІРУ ПС ПІД УПРАВЛІННЯМ ПЛАТФОРМИ WORDPRESS..... 73

ДОДАТОК В - ОПИС ПРОГРАМНОЇ СИСТЕМИ «INFOPLUG» РАННЬОГО ПРОГНОЗУВАННЯ РОЗМІРУ ПС ПІД УПРАВЛІННЯМ ПЛАТФОРМИ WORDPRESS..... 80

ДОДАТОК Г - ІНСТРУКЦІЯ КОРИСТУВАЧА ПРОГРАМНОЇ СИСТЕМИ «INFOPLUG» РАННЬОГО ПРОГНОЗУВАННЯ РОЗМІРУ ПС ПІД УПРАВЛІННЯМ ПЛАТФОРМИ WORDPRESS..... 81

ДОДАТОК Д - ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ ПРОГРАМНОЇ СИСТЕМИ «INFOPLUG» РАННЬОГО ПРОГНОЗУВАННЯ РОЗМІРУ ПС ПІД УПРАВЛІННЯМ ПЛАТФОРМИ WORDPRESS..... 82

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

ПЗ - програмне забезпечення;

ПК - персональний комп’ютер;

ПС - програмна система;

СКВ - система керування вмістом;

CL - the number of classes;

COCOMO - the COnstructive COst MOdel;

DIT - the depth of inheritance tree;

DM - the Mahalanobis distance;

KLOC - the thousand lines of code;

MBC - the number of methods by class;

MMRE - a mean magnitude of relative error;

MRE - a magnitude of relative error;

PHP - hypertext preprocessor;

PRED - percentage of prediction;

SEO - search engine optimization.

ВСТУП

Актуальність теми. В умовах стрімкого розвитку інформаційних технологій та зростання обсягів цифрового контенту особливої актуальності набуває розробка веб-систем і програмних комплексів, що функціонують на базі систем керування вмістом (CMS). Серед таких платформ провідне місце займає WordPress, яка забезпечує гнучкість, модульність та високу швидкість розробки веб-рішень різної складності - від персональних блогів до корпоративних інформаційних систем.

Розроблення та розвиток програмних систем під управлінням платформи WordPress часто потребує раннього прогнозування розміру майбутнього проєкту. Це дозволяє оцінити трудомісткість, вартість, обсяги ресурсів і строки реалізації. Традиційні методи оцінювання розміру програмного забезпечення не завжди забезпечують достатню точність у контексті специфіки WordPress, що зумовлює необхідність створення спеціалізованих математичних моделей раннього прогнозування [1].

Таким чином, розроблення математичної моделі для раннього прогнозування розміру програмних систем, створених під управлінням платформи WordPress, є актуальним завданням, а створення програми для її реалізації має вагоме практичне значення.

Мета і завдання дослідження. Метою роботи є підвищення достовірності раннього прогнозування розміру програмних систем, створених під управлінням платформи WordPress мовою програмування PHP.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- Проаналізувати існуючі моделі прогнозування розміру програмного забезпечення, зокрема для веб-систем, створених на платформі WordPress.
- Обґрунтувати необхідність побудови математичної моделі для раннього прогнозування розміру програмних систем під управлінням платформи WordPress.

- Побудувати нелінійну регресійну модель раннього прогнозування розміру програмних систем під управлінням платформи WordPress із використанням нормалізуючого перетворення.
- Розробити програму, що реалізує побудовану модель та забезпечує практичне прогнозування розміру відповідних програмних систем.

Об'єктом дослідження є процес раннього прогнозування розміру програмних систем, створених під управлінням платформи WordPress мовою програмування PHP.

Предметом дослідження є нелінійна регресійна модель раннього прогнозування розміру програмних систем, створених під управлінням платформи WordPress.

Методи дослідження. Для досягнення мети використано методи математичного моделювання, теорії ймовірності, математичної статистики, регресійного аналізу, об`єктно-орієнтованого програмування.

Наукова новизна полягає в удосконаленні математичної моделі раннього прогнозування розміру програмних систем під управлінням платформи Wordpress шляхом побудови нелінійної регресійної моделі з нормалізуючим перетворенням у вигляді десяткового логарифму. Це забезпечує підвищення достовірності раннього прогнозування розміру у порівнянні з існуючими моделями.

Практичне значення одержаних результатів. Розроблено програму, яка реалізує побудовану математичну модель раннього прогнозування розміру програмних систем під управлінням платформи WordPress. Створений інструмент може бути використаний для оцінювання трудомісткості, ресурсів і вартості майбутніх проєктів у галузі веб-розробки.

Особистий внесок здобувача. Кваліфікаційна робота є самостійно виконаною працею. Усі результати, подані у роботі, отримані автором особисто. У роботі здобувачеві належать: збір та аналіз емпіричних даних про метрики програмних систем під управлінням WordPress, попередня обробка даних для побудови нелінійної регресійної моделі раннього прогнозування їх розміру.

Апробація результатів досліджень. Основні положення та результати досліджень, викладені у роботі, були оприлюднені на VIII Всеукраїнській науково-практичній інтернет-конференції студентів, аспірантів та молодих вчених за тематикою «Сучасні комп’ютерні системи та мережі в управлінні» (24 листопада 2025 р., м. Херсон, м. Хмельницький) [2].

Публікації. За результатами досліджень опубліковано одну наукову працю, а саме матеріали конференції.

1 ДОСЛІДЖЕННЯ ІСНУЮЧИХ МЕТОДІВ ТА МОДЕЛЕЙ ДЛЯ РАННЬОГО ПРОНОЗУВАННЯ РОЗМІРУ ПРОГРАМНИХ СИСТЕМ ПІД УПРАВЛІННЯМ ПЛАТФОРМИ WORDPRESS

1.1 Особливості та переваги використання WordPress як платформи керування вмістом веб-сайтів

Системи керування вмістом (CMS, Content Management System) є потужним інструментом для швидкого та ефективного створення, редагування та підтримки веб-сайтів без необхідності глибоких технічних знань. Веб-розробники використовують CMS для оптимізації процесу розробки, автоматизації стандартних функцій і зручного управління контентом. Однією з найбільш популярних CMS є WordPress, яка користується високим попитом завдяки своїм функціональним можливостям, простоті використання і великій спільноті підтримки.

На сьогодні WordPress займає велику частину всіх веб-сайтів у світі, що робить його найбільш популярною платформою для створення інтернет-ресурсів. Серед користувачів WordPress - як маленькі блоги та особисті сайти, так і великі медіа-компанії та урядові установи. Відомі ресурси, такі як Forbes, Vogue, The New York Times, а також наукові організації, зокрема NASA та Білий Дім, активно використовують WordPress для своїх онлайн-ресурсів. Це підкреслює надійність і універсальність платформи, що відповідає потребам різноманітних типів сайтів.

Основні функції WordPress

Створення та редагування контенту. WordPress дозволяє користувачам легко створювати та редагувати сторінки, пости та інші види контенту за допомогою інтуїтивно зрозумілого інтерфейсу.

Візуальний редактор Gutenberg. Це потужний текстовий редактор, що дозволяє працювати з контентом у двох режимах: візуальному та HTML. Це

забезпечує більшу гнучкість при створенні та редагуванні контенту, зокрема для новачків.

Гнучкість у налаштуваннях URL-адрес. Користувачі можуть легко налаштувати структуру URL своїх сторінок і постів, що позитивно впливає на SEO (оптимізацію для пошукових систем).

Інтеграція медіа-контенту. В WordPress можна без проблем додавати зображення, відео та інші медіа-ресурси, а також здійснювати їх редагування прямо в адмін-панелі.

Розширення можливостей через плагіни. Плагіни - це додаткові модулі, написані на PHP, що інтегруються в систему і додають нові функції. Вони дають змогу адаптувати сайт під конкретні потреби без необхідності змінювати основний код системи.

З офіційного каталогу WordPress доступно більше 57000 безкоштовних плагінів, що охоплюють різні аспекти роботи сайту, від SEO-оптимізації до забезпечення безпеки та аналітики. Плагіни дозволяють значно розширити функціональність платформи, не вимагаючи програмування.

Популярні плагіни включають інструменти для управління формами зворотного зв'язку, інтеграції з соціальними мережами, оптимізації швидкості завантаження сайту та багато інших.

WordPress дозволяє професійним розробникам змінювати поведінку системи за допомогою плагінів та кастомних тем. Розробники, що володіють мовами програмування, зокрема PHP, можуть створювати складні функціональні рішення або адаптувати існуючі плагіни, що до звичайних користувачів WordPress пропонує простий та інтуїтивно зрозумілий інтерфейс, що не вимагає від них глибоких знань у веб-розробці. Це робить платформу доступною для широкого кола людей.

WordPress підходить для створення будь-яких типів сайтів, від персональних блогів до корпоративних порталів, інтернет-магазинів та великих медіа-ресурсів. Це забезпечує широку можливість масштабування, при цьому користувачі можуть збільшувати функціональність своїх сайтів за допомогою плагінів і тем [3].

1.2 Дослідження існуючих методів та моделей для раннього прогнозування розміру програмних систем

Оцінювання розміру програмних систем (ПС) є одним з ключових етапів у процесі програмної інженерії, оскільки воно безпосередньо впливає на планування строків розробки, розподіл ресурсів, визначення вартості та оцінку ризиків проєкту. На ранніх стадіях життєвого циклу ПС така оцінка має прогнозний характер і використовується для прийняття управлінських рішень. Для систем, створених під управлінням WordPress, оцінювання розміру має свої особливості, оскільки платформа ґрунтується на модульній архітектурі, що включає ядро, теми та плагіни. Саме тому стандартні моделі оцінювання, розроблені для традиційних програмних систем, не завжди забезпечують достатню достовірність при застосуванні до екосистеми WordPress.

Метрика програмного забезпечення - це величина, що дозволяє отримати числове значення певних характеристик та параметрів програмної системи, а також оцінити окремі властивості частини програмного коду. Кожна метрика зазвичай має еталонні значення, що визначають порогові значення, за яких слід звернути увагу на конкретну ділянку коду.

Метрики коду класифікуються на різні категорії і дозволяють оцінювати різні аспекти програмної системи: складність і організацію коду, зв'язність компонентів, обсяг програмних елементів і так далі. Наприклад, метрика кількості рядків коду є однією з найпростіших для розуміння і обчислення. У поєднанні з іншими метриками вона може забезпечити формалізовані дані для оцінювання коду. Таким чином, можна побудувати зв'язок між кількістю рядків коду в класі і числом методів у цьому класі, що дозволить визначити об'ємність методів класу. Окрім того, ці оцінки можна використовувати разом з метриками складності для виявлення найбільш складних ділянок програмного коду. Зазвичай метрики складності важко обчислити на ранніх етапах проєкту, оскільки для їх розрахунку

потрібне більш детальне проєктування. Однак вони є важливими для отримання прогнозних оцінок щодо тривалості та вартості розробки програмних систем, оскільки визначають їх трудомісткість.

На даний момент методи оцінювання розміру ПС можна поділити на три основні групи, а саме:

- емпіричні методи;
- функціональні методи;
- математичні (регресійні) моделі.

Розглянемо більш детальніше основні моделі з яких, однією з найвідоміших є COCOMO (Constructive COst MOdel), конструктивна вартісна модель регресії, яка базується на кількості рядків коду (LOC - Lines of Code) та великому досвіді реалізації програмних проєктів. Модель була розроблена на основі збору даних із багатьох програмних проєктів і аналізу цієї інформації, що дозволило створити ефективні формули для оцінювання витрат.

Серед функціональних методів, котрі спираються на кількість функціональних можливостей, які реалізує система, незалежно від мови програмування, найбільш поширеним представником є метод функціональних точок (Function Point Analysis, FPA), а також його модифікації: COSMIC FFP, Mark II FPA, NESMA FPA.

Розглянемо регресійні моделі оцінювання розміру ПС, зокрема лінійні та нелінійні. У лінійних моделях, використовуючи емпіричні дані, за допомогою методу найменших квадратів визначають коефіцієнти рівняння регресії. Це рівняння дозволяє оцінити розмір ПС. У випадку з нелінійними моделями існують два підходи: перший передбачає перебір нелінійної функції регресії, на основі якої будується модель, а другий полягає в знаходженні попередніх перетворень випадкових змінних (нормалізуючих перетворень), що забезпечують лінійний зв'язок між уже нормалізованими змінними.

Зазвичай для побудови нелінійних моделей використовується десятковий логарифм як нормалізуюче перетворення. Наприклад, моделі, такі як COCOMO, COCOMO II, ISBSG, були створені з використанням цього перетворення у вигляді

десяткового логарифму. Отже, вдосконалення математичної моделі для раннього прогнозування розміру програмних систем під управлінням платформи Wordpress мовою програмування PHP, буде здійснюватися також на основі нормалізуючого перетворення у вигляді десяткового логарифму.

1.3 Обгрунтування необхідності проведення досліджень за обраною темою

У сучасному світі, де розробка веб-систем і програмних продуктів взагалі займає важливе місце в економічному і технологічному прогресі, достовірне раннє прогнозування розміру програмних систем стає критично важливим для успішного планування, управління ресурсами та забезпечення якості. Платформи систем керування вмістом, такі як WordPress, є одними з найпоширеніших і використовуються для створення різноманітних веб-сайтів, від особистих блогів до великих корпоративних порталів. Проте, через свою модульну архітектуру та безліч плагінів і тем, раннє прогнозування розміру таких систем є значно складнішим порівняно з традиційними програмними продуктами [3].

Одним з основних аспектів у розробці веб-систем є якісне прогнозування розміру програмних систем на ранніх етапах проєкту. Це дозволяє здійснити більш точні прогнози щодо затрат часу, трудомісткості, вартості проєкту, а також для визначення потенційних проблем і ризиків. Поширення платформ, таких як WordPress, на фоні зростання використання плагінів, що значно змінюють функціональність сайту, потребує удосконалення моделей для оцінювання їх розміру та трудомісткості.

На сьогоднішній день більшість існуючих моделей оцінювання розміру ПС не повною мірою підходять для модульних систем, як WordPress, через їх складність та взаємозалежність між плагінами, темами, базами даних та інтерфейсами. Нелінійні регресійні моделі, які зараз активно використовуються в багатьох методах оцінювання, здебільшого не враховують специфіку веб-розробки та особливості таких платформ, як WordPress, що значно знижує їх достовірність.

Тому проведення дослідження в рамках створення математичної моделі раннього прогнозування розміру програмних систем під управлінням платформи WordPress є необхідним для покращення достовірності оцінки розміру, зменшення ризиків, оптимізації витрат і часу на розробку та поліпшення якості програмних продуктів в екосистемі WordPress.

1.4 Висновки за розділом 1

У межах розділу було сформовано теоретичне підґрунтя для подальшого дослідження, показано актуальність використання WordPress як однієї з найпоширеніших платформ керування вмістом та підкреслено її особливості, що впливають на процес раннього оцінювання розміру програмних систем. Подальший аналіз існуючих методів і моделей прогнозування дозволив виявити їх сильні сторони та обмеження, особливо в контексті модульних систем, де структура розробки значною мірою визначається взаємодією плагінів, тем і ядра платформи.

Встановлено, що стандартні підходи, попри їхню поширеність та перевіреність, не завжди забезпечують достатню достовірність оцінок для систем, побудованих під WordPress, оскільки такі системи характеризуються складною внутрішньою архітектурою та високою варіативністю функціональних елементів. Це зумовлює потребу у створенні більш адаптованої моделі, здатної враховувати специфіку метрик плагінів і структурних особливостей веб-систем, створених на PHP для WordPress.

Таким чином, проведений огляд підтвердив необхідність розробки вдосконаленої нелінійної регресійної моделі раннього прогнозування розміру відповідних програмних систем. Отримані результати стали основою для подальших етапів дослідження, де буде здійснено збір емпіричних даних, побудову моделі та оцінювання її якості.

2 РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ ДЛЯ РАННЬОГО ПРОГНОЗУВАННЯ РОЗМІРУ ПРОГРАМНИХ СИСТЕМ ПІД УПРАВЛІННЯМ ПЛАТФОРМИ WORDPRESS

2.1 Аналіз наявних математичних моделей та вибір факторів для раннього прогнозування розміру програмних систем під управлінням платформи WordPress

Раннє прогнозування розміру програмних систем є однією з ключових задач інженерії програмного забезпечення, оскільки від точності оцінки обсягу коду залежать планування ресурсів, трудомісткість розробки та вартість проєкту. Для програмних систем під управлінням платформи WordPress ця задача ускладнюється високою варіативністю архітектури програмної системи, відмінностями у стилі програмування та активним використанням об’єктно-орієнтованих конструкцій мови PHP.

Серед наявних математичних моделей прогнозування розміру програмних систем можна виділити кілька основних груп. До першої групи належать аналогові моделі, які базуються на порівнянні нового проєкту з раніше реалізованими системами. Хоча такі моделі є простими у використанні, їх точність значною мірою залежить від якості історичних даних і суб’єктивності вибору аналогів, що обмежує можливість їх застосування для програмних систем WordPress з унікальною структурою.

Друга група представлена експертними моделями, у яких оцінювання здійснюється на основі досвіду фахівців. Основним недоліком цього підходу є високий рівень суб’єктивності та складність формалізації знань експертів, що знижує відтворюваність результатів.

До третьої групи належать алгоритмічні моделі, зокрема лінійні та нелінійні регресійні моделі, які встановлюють формальний зв’язок між розміром програмної системи та її кількісними характеристиками. Лінійні моделі є простими в реалізації та інтерпретації, однак часто не здатні адекватно

відобразити складні нелінійні залежності між метриками програмного коду. Це особливо характерно для програмних ситем WordPress, де вплив архітектурних факторів на розмір системи є мультиплікативним.

Окрему групу становлять моделі на основі машинного навчання, такі як нейронні мережі або дерева рішень. Вони здатні досягати високої точності прогнозування, однак потребують великих навчальних вибірок, мають складну інтерпретацію та є обчислювально витратними, що ускладнює їх застосування на ранньому етапі проєктування.

З урахуванням зазначених особливостей, доцільним є використання нелінійної регресійної моделі, яка поєднує відносну простоту реалізації з можливістю врахування нелінійних залежностей між змінними. Такий підхід дозволяє формалізувати зв’язок між розміром програмної системи та її структурними метриками, а також забезпечує можливість статистичного оцінювання точності прогнозу та побудови довірчих інтервалів і інтервалів прогнозування.

Вибір факторів для раннього прогнозування розміру програмних систем під управлінням платформи WordPress ґрунтується на доступності метрик на початкових етапах розроблення та їхній здатності адекватно відображати складність програмної архітектури. До таких факторів віднесено кількість класів, середню кількість методів у класі та глибину дерева успадкування. Ці показники є типовими об’єктно-орієнтованими метриками, які можуть бути отримані ще до повної реалізації програмного коду та мають безпосередній вплив на підсумковий розмір системи, виражений у тисячах рядків коду.

2.2 Попередня обробка емпіричних даних для раннього прогнозування розміру програмних систем під управлінням платформи WordPress

Для попередньої обробки емпіричних даних раннього прогнозування розміру програмних систем під управлінням платформи WordPress [4], були зібрані дані з метрик 109 програмних систем:

- найменування програмних систем;
- розмір програмних систем у тисячних рядків коду KLOC;
- кількість класів CL;
- середня кількість методів на клас MBC;
- глибина дерева успадкування DIT.

Фрагмент зібраних метрик подано у таблиці 2.1.

Таблиця 2.1 - Фрагмент метрик програмних систем разом із квадартом відстані Махаланобіса D_M

№	Name	KLOC	Classes	MBC	DIT	D_M
1	2	3	4	5	6	7
1	wp-forms-pro	12,834	145	7,21	1,33	14,707
2	seo-optimizer-lite	8,219	87	9,45	1,22	4,6882
3	simple-login-security	2,115	18	11,07	1,10	3,0833
4	social-media-connect	5,943	64	8,83	1,26	4,7749
5	cache-master	34,621	272	6,41	1,09	13,683
6	image-compressor-pro	15,983	82	9,92	1,40	9,3267
7	contact-form-builder	7,612	43	10,37	1,17	0,8584
8	wp-backup-manager	27,540	205	5,84	1,29	5,8293
9	analytics-dashboard-lite	19,124	98	7,56	1,35	3,6832
10	translate-plus	4,812	26	12,48	1,18	5,9816
11	e-commerce-kit	82,530	615	8,02	1,23	8,2918
12	security-enhancer	43,277	354	9,11	1,30	7,5453
13	gallery-showcase	9,153	47	10,01	1,21	0,5648
14	mail-delivery-system	11,505	53	7,64	1,27	1,8638

Продовження таблиці 2.1

15	site-optimizer-pro	21,893	174	6,52	1,31	5,7359
16	popup-manager-lite	5,601	22	11,44	1,15	2,4314
17	woocommerce-advanced	136,443	998	8,36	1,28	25,929
18	event-calendar-plus	68,204	417	7,73	1,35	13,483
19	shortcode-collection	3,418	12	9,88	1,10	2,2767
20	spam-protector	2,907	8	13,16	1,06	7,7704
21	wp-gallery-pro	16,243	92	10,75	1,21	1,5669
22	login-guard	2,635	10	9,41	1,18	0,8271
23	media-uploader	9,817	55	8,13	1,28	1,4954
24	wp-cloud-sync	18,301	111	6,98	1,32	2,9409
25	content-duplicator	4,507	25	11,56	1,10	3,2441
26	wp-cache-engine	22,415	138	7,22	1,25	1,9905
27	woocommerce-tools	71,382	612	9,08	1,29	24,444
28	elementor-style-pack	33,821	179	10,45	1,36	10,49
29	api-manager	6,344	33	7,87	1,11	5,1876
30	wp-translate-tools	9,503	49	12,02	1,16	3,8697
31	theme-customizer	13,290	71	8,55	1,34	3,147
32	security-shield	47,832	377	8,92	1,28	4,8622
33	image-gallery-pro	5,912	21	10,63	1,19	1,2246
34	form-builder-lite	7,463	38	9,21	1,12	2,0356
35	woocommerce-payments	119,734	932	8,84	1,27	24,667
36	wp-firewall-pro	41,236	290	9,03	1,40	6,4951
37	backup-restore	27,188	201	6,51	1,26	3,9006
38	mailchimp-integration	8,765	48	8,73	1,21	0,5044
39	analytics-tools	18,623	97	6,84	1,35	4,3721
40	wp-site-inspector	10,478	59	9,64	1,20	5,7359

У таблиці 2.1 також одразу вирахувані значення квадрату відстані Махаланобіса D_M [5]. Значення D_M для кожної багатовимірної точки даних i, i = 1,2, …,N, визначається як:

$$D_M = (X_i - X)^T S_N^{-1} (X_i - X) \tag{2.1}$$

де X_i - випадковий вектор $X = \{X1,X2,\ldots,Xm\}^T$ для кожної багатовимірної точки даних i, $i = 1,2,\ldots,N$;

m - кількість компонентів вектору X, в нашому випадку дорівнює 4,

$\bar{X}$ - вектор вибіркових середніх;

S_N - вибіркова коваріаційна матриця:

$$S_N = \frac{1}{N}\sum_{i=1}^{N}(X_i - X)(X_i - X)^T. \quad (2.2)$$

Про негаусівський розподіл чотиривимірних даних свідчить значення багатовимірного ексцесу β_2, оцінка якого визначається завдяки тесту Мардіа, за наступною формулою:

$$\beta_2 = \frac{1}{N}\sum_{i-1}^{N}\{(X_i - X)^T S_N^{-1}(X_i - X)\}^2 \quad (2.3)$$

Відомо, що для багатовимірного нормального розподілу $\beta_2 = m(m+2)$. У нашому випадку при m = 4, $\beta_2 = 24$. Для чотиривимірних даних з таблиці 2.1 оцінка β_2 за формулою 2.3 дорівнює 38,681.

Зважаючи на негаусівський розподіл чотиривимірних даних, для виявлення викидів ми використовуємо метод, заснований на нормалізуючих перетвореннях та на квадраті відстані Махаланобіса D_M, що для кожної багатовимірної точки нормалізованих даних i, $i = 1,2,\ldots,N$, визначається як

$$d_i^2 = (Z_i - Z)^T S_N^{-1}(Z_i - Z) \quad (2.4)$$

де $\bar{Z}$ - вектор вибіркових середніх нормалізованих даних;

S_N - вибіркова коваріаційна матриця нормалізованих даних.

$$S_N = \frac{1}{N}\sum_{i=1}^{N}(Z_i - Z)(Z_i - Z)^T \quad (2.5)$$

де Z_i - гаусівський випадковий вектор $Z = \{Z1,Z2,\ldots,Zm\}T$ для кожної багатовимірної точки даних i, $i = 1,2,\ldots,N$; m - кількість компонентів вектору Z.

Вектор Z в загальному випадку отримують за допомогою багатовимірного нормалізуючого перетворення негаусівського випадкового вектору, яке має вигляд:

$$X = \{X1,X2,\ldots,Xm\}^T \tag{2.6}$$

У цьому разі, в перетворенні (2.6) $\psi = \{lgY\ ,lgX1\ ,lgX2\ ,lgX3\}^T$.

Якщо дані мають багатовимірний нормальний розподіл, то розподіл DM поводиться як розподіл $\chi 2$. Для великих N - m (принаймні більше ніж 25), відстань d_i^2 повинна вести себе як незалежні випадкові величини $\chi_{m,\alpha\text{-}2}$, де $\chi_{m,\alpha\text{-}2}$ - це квантиль розподілу χ^2 з рівнем значимості α. Також, якщо дані мають багатовимірний нормальний розподіл, то статистика

$$\frac{N(N-m)d_i^2}{m(N^2-1)} \tag{2.7}$$

поводиться як F-розподіл. Також статистика (2.4) повинна вести себе приблизно як незалежні випадкові величини $F_{m,N-m,\alpha}$, тут $F_{m,N-m,\alpha}$ - це квантиль F-розподілу з рівнем значимості α.

Точка даних, для якого величина d_i^2 більша, ніж квантиль розподілу χ^2, та величина статистики (2.7) більша, ніж квантиль F-розподілу, розглядається як багатомірний викид та відкидається. Після відкидання зменшене число багатомірних точок даних знову нормалізується, використовуючи перетворення:

$$T = \psi(P), \tag{2.8}$$

поки всі рядки даних не будуть мати величину d_i^2 не більшу, ніж квантиль розподілу χ^2, або величину статистики (2.7) не більшу, ніж квантиль F-розподілу.

Надалі зібрані чотиривимірні негаусові дані використовуються для побудови нелінійної регресійної моделі для раннього прогнозування розміру програмних систем, що працюють під управлінням платформи WordPress .

Спершу потрібно одержати лінійну регресійну модель для оцінювання розміру програмних систем, створених для системи керування вмістом WordPress на PHP, яка представляється у вигляді:

$$Y = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + \varepsilon \quad (2.9)$$

де оцінки параметрів b_0, b_1, b_2, b_3 знаходяться методом найменших квадратів;

ε - випадкова величина з розподілом Гаусу, $\varepsilon \sim N(0,\sigma^2_\varepsilon)$.

Тепер потрібно перевірити нульову гіпотезу про нормальність закону розподілу випадкової величини ε за критерієм Пірсона для моделі (2.9).

У випадку великої вибірки значень випадкової величини, перевірку нормальності закону розподілу результатів виконують за критерієм Пірсона (критерієм χ^2). Відповідно за цим критерієм спочатку обчислюють значення χ^2:

$$\chi^2 = \sum_{j=1}^{m} \frac{(n_j - np_j)^2}{np_j}, \quad (2.10)$$

де m - кількість підінтервалів, на які розбивається інтервал $[x_{min}, x_{max}]$;

x_{min} і x_{max} - мінімальне і максимальне значення випадкової величини X;

n_j - абсолютна частота в j-ому підінтервалі (кількість значень випадкової величини у j-ому підінтервалі);

p_j - ймовірність попадання значень випадкової величини X у j-ий підінтервал.

Кількість підінтервалів, на які розбивається інтервал $[x_{min}, x_{max}]$ може бути визначений за формулами або Старджеса: $m = log2n$ +1 = 3,3lgn + 1 або Брукса і Карузера: $m = 5lgn$. Було використано формулу Старджеса, за якою m = 8.

Ймовірність попадання значення випадкової величини X у j-ий підінтервал може бути визначена за формулою:

$$p_j = \int_{x_{j-1}}^{x_j} f(x)dx, \tag{2.11}$$

де $x_j - 1$ і x_j - ліва і права границі j-го підінтервалу;

f(x) - функція щільності ймовірності нормального розподілу (розподілу Гауса):

$$f(x) = \frac{1}{\sigma_x\sqrt{2\pi}} e^{-\frac{(x-m_x)^2}{2\sigma_x^2}} \tag{2.12}$$

де m_x - математичне сподівання випадкової величини X.

Потім, в залежності від рівня значимості α та кількості ступенів вільності ν за таблицею верхніх 100α %-вих точок розподілу χ^2 знаходять значення $\chi^2_{кр}$. Кількість ступенів вільності визначається як $\nu = m-k-1$, де k - кількість параметрів, від яких залежить закон розподілу. Для нормального розподілу $k = 2$.

Якщо $\chi^2 \leq \chi^2_{кр}$, то з ймовірністю $1-\alpha$ може бути прийнята гіпотеза про те, що закон розподілу результатів спостережень є нормальним, а якщо $\chi^2 > \chi^2_{кр}$ - цю гіпотезу потрібно відкинути.

Побудуємо лінійну регресійну модель на основі вибірки, що складається зі 109 експериментальних точок. Оцінивши параметри моделі відповідно до формули (2.9) із використанням методу найменших квадратів. У результаті отримаємо такі значення коефіцієнтів лінійної регресії:

b_0 = -12,729225, b_1 = 0,129795, b_2 = 0,108727, b_3 = 12,25484.

Для перевірки адекватності побудованої моделі проведемо аналіз залишків із застосуванням критерію χ^2 Пірсона відповідно до формули (2.10). За результатами розрахунків отримано значення статистики χ^2 = 63,98, яке перевищує критичне значення $\chi^2_{кр}$ = 11,07.

Це вказує на відсутність теоретичного обґрунтування для використання моделі лінійної регресії для раннього прогнозування розміру програмних систем під управлінням платформи WordPress та зумовлює необхідність побудови нелінійної регресійної моделі [6].

2.3 Побудова нелінійної регресійної моделі для раннього прогнозування розміру програмних систем під управлінням платформи WordPress

Для побудови нелінійної регресійної моделі для раннього прогнозування розміру програмних систем під управлінням платформи WordPress, було сформовано початковий набір даних метрик зі 109 програмних систем.

Для кожної програмної системи використовуються такі метрики: кількість рядків програмного коду у тисячах (KLOC), кількість класів (Classes), середня кількість методів у класі (MBC) та глибина дерева успадкування (DIT).

Зазначені метрики, окрім KLOC, використовуються як незалежні змінні для побудови нелінійної регресійної моделі для раннього прогнозування розміру програмних систем під управлінням платформи WordPress.

На першому етапі було виконано перевірку змінних на мультиколінеарність, за результатами отримано таку обернену коваріаційну матрицю, елементи якої мають значення

$$S_N = \begin{matrix} 1{,}218 & 0{,}192 & -0{,}387 \\ 0{,}192 & 1{,}415 & 0{,}669 \\ -0{,}387 & 0{,}669 & 1{,}508 \end{matrix}.$$

Наступним кроком було порівняння значень елементів головної діагоналі матриці з критичним значенням, що дорівнює 5, отриманий результат свідчить про відсутність критичної мультиколінеарності, що дозволило використати всі обрані метрики при побудові багатофакторної регресійної моделі.

При аналізі багатовимірного розподілу даних за критеріями Мардіа [7] отримали наступні значення: статистика z_1 = 268,9625 при критичному квантилі b_1 39,9968, значення багатовимірного ексцесу b_2 = 38,6810 при відповідному критичному квантилі 28,5372.

Отримані результати вказують на істотне відхилення даних від багатовимірного нормального розподілу, що обґрунтовує застосування десяткового логарифмування змінних, після чого нелінійна регресійна модель має вигляд:

$$Y = 10^{\varepsilon + b_0} X_1^{b_1} X_2^{b_2} X_3^{b_3} \quad (2.13)$$

Після виконання логарифмічного перетворення (2.13) було проведено ітеративну перевірку на наявність викидів із використанням квадрата відстані Махаланобіса згіно з [8] та критерію χ^2 з критичним значенням 14,8603.

Фрагмент нормалізованих даних подано у таблиці 2.2.

Таблиця 2.2 - Фрагмент нормалізованих даних

№	Name	KLOC	Classes	MBC	DIT	D_i^2
1	2	3	4	5	6	7
1	wp-forms-pro	1,108	2,161	0,858	0,124	15,506
2	seo-optimizer-lite	0,915	1,940	0,975	0,086	14,742
3	simple-login-security	0,325	1,255	1,044	0,041	18,391
4	social-media-connect	0,774	1,806	0,946	0,100	18,187
5	cache-master	1,539	2,435	0,807	0,037	17,651
6	image-compressor-pro	1,204	1,914	0,997	0,146	8,975
7	contact-form-builder	0,881	1,633	1,016	0,068	1,167
8	wp-backup-manager	1,440	2,312	0,766	0,111	7,430
9	analytics-dashboard-lite	1,282	1,991	0,879	0,130	4,423
10	translate-plus	0,682	1,415	1,096	0,072	5,351
11	e-commerce-kit	1,917	2,789	0,904	0,090	6,241
12	security-enhancer	1,636	2,549	0,960	0,114	4,251
13	gallery-showcase	0,962	1,672	1,000	0,083	0,573
14	mail-delivery-system	1,061	1,724	0,883	0,104	3,829

Продовження таблиці 2.2

15	site-optimizer-pro	1,340	2,241	0,814	0,117	5,233
16	popup-manager-lite	0,748	1,342	1,058	0,061	3,131
17	woocommerce-advanced	2,135	2,999	0,922	0,107	8,640
18	event-calendar-plus	1,834	2,620	0,888	0,130	4,134
19	shortcode-collection	0,534	1,079	0,995	0,041	5,320
20	spam-protector	0,463	0,903	1,119	0,025	11,138
21	wp-gallery-pro	1,211	1,964	1,031	0,083	2,074
22	login-guard	0,421	1,000	0,974	0,072	5,717
23	media-uploader	0,992	1,740	0,910	0,107	2,110
24	wp-cloud-sync	1,262	2,045	0,844	0,121	3,489
25	content-duplicator	0,654	1,398	1,063	0,041	4,136
26	wp-cache-engine	1,351	2,140	0,859	0,097	1,899
27	woocommerce-tools	1,854	2,787	0,958	0,111	6,465
28	elementor-style-pack	1,529	2,253	1,019	0,134	7,718
29	api-manager	0,802	1,519	0,896	0,045	5,368
30	wp-translate-tools	0,978	1,690	1,080	0,064	3,567
31	theme-customizer	1,124	1,851	0,932	0,127	3,596
32	security-shield	1,680	2,576	0,950	0,107	3,611
33	image-gallery-pro	0,772	1,322	1,027	0,076	4,425
34	form-builder-lite	0,873	1,580	0,964	0,049	1,906
35	woocommerce-payments	2,078	2,969	0,946	0,104	8,769
36	wp-firewall-pro	1,615	2,462	0,956	0,146	5,920
37	backup-restore	1,434	2,303	0,814	0,100	4,345
38	mailchimp-integration	0,943	1,681	0,941	0,083	0,547
39	analytics-tools	1,270	1,987	0,835	0,130	6,088
40	wp-site-inspector	1,020	1,771	0,984	0,079	0,254

На першій ітерації максимальне значення квадрата відстані Махаланобіса d_i^2 становило 18,3915, у результаті чого було виявлено один викид із номером 3.

На другій ітерації максимальне значення d_i^2 дорівнювало 21,2764, що призвело до вилучення точки даних з номером 4.

На третій ітерації значення d_i^2 max склало 21,8407, ідентифіковано викид із номером 2.

На четвертій ітерації максимальне значення d_i^2 становило 26,2144, що відповідає викиду з номером 1.

На п’ятій ітерації d_i^2 max становило 17,1858, унаслідок чого було вилучено точку даних з номером 5.

На шостій ітерації максимальне значення d_i^2 дорівнювало 15,3031, що перевищує критичний квантиль, тому була вилучена точка даних з номером 20.

На наступній ітерації, при значенні d_i^2 max = 9,7563, яке не перевищує критичне значення 14,8603, викиди за критерієм Махаланобіса не виявилися. Водночас, за результатами аналізу інтервалу передбачення нелінійної регресії було знайдено п’ять точок даних, що виходять за його межі, а саме, проєкти з номерами 15, 33, 63, 66 та 88. Вони також були видалені з набору даних.

Таким чином, загальна кількість вилучених викидів становить 11 точок даних із номерами 3, 4, 2, 1, 5, 20, 15, 33, 63, 66 та 88, після чого для подальшого аналізу залишається 98 коректних записів із початкових 109.

Було прийнято рішення, очищений набір даних поділити на дві частини: 60 проєктів використано для побудови нелінійної регресійної моделі, а 38 - для тестування її якості. Для навчальної вибірки побудовано лінійну регресійну модель, для якої отримано такі оцінки параметрів: $b_0 = -0{,}4866$, $b_1 = 0{,}8414$, $b_2 = 0{,}0171$, $b_3 = 0{,}2789$.

Для цієї ж вибірки було побудовано нелінійну регресійну модель, яка має вигляд:

$$Y = 10^{\varepsilon - 0{,}4866} X_1^{0{,}8414} X_2^{0{,}0171} X_3^{0{,}2789}, \qquad (2.14)$$

яка продемонструвала високі показники якості: значення коефіцієнта детермінації $R^2 = 0{,}9836$, та скоригованого коефіцієнта детермінації R^2_adj = 0,9827. Середня відносна похибка MMRE становить 0,0800, а показник Pred(0,25) дорівнює 1, що відповідає 60 коректним прогнозам із 60 можливих.

Оцінювання якості моделі на тестовій вибірці з 38 спостережень підтвердило її високу якість. Для цієї вибірки значення коефіцієнта детермінації

R^2 становить 0,9849, середнє значення відносної похибки MMRE дорівнює 0,0695, а показник Pred(0,25) також становить 1,0000, що означає, що всі прогнозні значення потрапляють у межі допустимої похибки [9, 10].

2.4 Висновки за розділом 2

У розділі виконано перевірку емпіричних даних на наявність викидів для побудови нелінійної регресійної моделі раннього прогнозування розміру програмних систем, що працюють під управлінням платформи WordPress, а також сформовано таку модель для раннього прогнозування розміру цих програмних систем.

1. Перевірку даних, призначених для побудови нелінійної регресійної моделі раннього прогнозування розміру програмних систем під управлінням WordPress, здійснено за методом, який ґрунтується на нормалізуючих перетвореннях та використанні квадрата відстані Махаланобіса.
2. На основі застосування нормалізуючого перетворення у вигляді десяткового логарифму було удосконалено регресійну модель для раннього прогнозування розміру програмних систем під управлінням WordPress.
3. Побудована нелінійна регресійна модель для раннього прогнозування розміру програмних систем під управлінням платформи WordPress демонструє прийнятну якість оцінювання.

Результати досліджень, викладених у цьому розділі, опубліковано автором у роботі [2].

3 РОЗРОБКА ПРОЄКТНИХ РІШЕНЬ ДЛЯ ПРОГРАМНОЇ СИСТЕМИ РАННЬОГО ПРОГНОЗУВАННЯ РОЗМІРУ ПС ПІД УПРАВЛІННЯМ ПЛАТФОРМИ WORDPRESS

3.1 Постановка задачі на розробку програмної системи раннього прогнозування розміру ПС під управлінням платформи WordPress

Виходячи із попереднього аналізу предметної області та побудови математичної моделі, виконаємо постановку задачі на розробку програмної системи раннього прогнозування розміру ПС під управлінням платформи WordPress.

Програмна система призначена для раннього прогнозування розміру ПС під управлінням платформи WordPress.

Програма має виконувати такі функції:

1) Вводити з інтерфейсу програми три метрики діаграми класів, за якими потрібно оцінити розмір ПС, створених для системи керування вмістом WordPress на мові програмування PHP в тисячах строк коду: друга , третя 1 X 2 та четверта X 3 метрики визначають відповідно загальну кількість класів, середню кількість методів на клас та глибину дерева успадкування.

2) Виконувати обчислення раннього прогнозування розміру програмних систем під управлінням платформи WordPress.

3) Виконувати обчислення довірчого інтервалу нелінійної регресії.

4) Виконувати обчислення інтервалу передбачення нелінійної регресії.

Вимоги до організації вхідних та вихідних даних

Вхідні дані:

- три метрики діаграми класів, за якими потрібно для раннього прогнозування розміру програмних систем під управлінням платформи WordPress: загальна кількість класів, кількість методів на клас та глибина дерева успадкування.

Вихідні дані:

- значення границь довірчого інтервалу нелінійної регресії та границь інтервалу передбачення нелінійної регресії раннього прогнозування розміру програмних систем під управлінням платформи WordPress.

Програма має експлуатуватись на обладнанні з параметрами не нижчими за наступні: 64-бітний процесор на архітектурі x86 або ARM, оперативна пам’ять 2 Гб, простір на жорсткому диску 800 Мб.

Для запуску та роботи з програмним забезпеченням необхідна операційна система Windows 7/8/10/11.

3.2 Загальносистемні рішення для програмної системи раннього прогнозування розміру ПС під управлінням платформи WordPress

3.2.1 Вибір засобів моделювання для ПС прогнозування розміру

Для даного проєкту застосування UML є найбільш доцільним, оскільки програмна система прогнозування розміру має чітко виражену структурну та обчислювальну логіку, що включає введення метрик, математичну модель прогнозування та формування результатів. UML дозволяє формально відокремити рівень інтерфейсу користувача, логіку обчислень і зберігання параметрів моделі, що підвищує зрозумілість архітектури системи.

Крім того, UML добре підходить для опису систем, що базуються на математичних моделях, оскільки дає змогу наочно представити зв’язки між сутностями, такими як метрики програмної системи, модель прогнозування та результат оцінювання. Використання діаграм класів і діаграм взаємодії забезпечує коректне відображення як статичної структури системи, так і динаміки її роботи під час виконання прогнозу.

Таким чином, застосування UML у даному проєкті дозволяє створити цілісну, формалізовану та зрозумілу модель для раннього прогнозування розміру

ПС під управлінням платформи Wordpress, що сприяє підвищенню якості проєктування, спрощує реалізацію та забезпечує відповідність сучасним вимогам програмної інженерії [11].

3.2.2 Розробка діаграми варіантів використання ПС прогнозування розміру

Проєкт програми раннього прогнозування розміру ПС під управлінням платформи Wordpress передбачає наявність одного актора - користувача програми, який зможе використовувати всі функції створеної програми, такі, як вводити дані метрик, виконувати розрахунок розміру ПС, а також обчислювати довірчий інтервал і інтервал передбачення для нелінійної регресії [11].

В таблиці 3.1 представлений опис виявлених варіантів використання ПС прогнозування розміру.

Таблиця 3.1 - Виявлені варіанти використання

Актори	Найменування	Формулювання
Користувач	Ввести дані метрик	Вводить метрики діаграми класів, за якими буде виконуватись прогнозування розміру програмних систем: загальну кількість класів, загальну кількість зв’язків між класами та загальну кількість методів.
Користувач	Ввести параметри моделі	Вводить параметрі моделі, за якими буде виконуватись побудова моделі прогнозування розміру програмних систем
Користувач	Отримати прогноз	Отримує розмір ПС прогнозування розміру програмних систем у тисячах рядків коду
Користувач	Отримати довірчий інтервал нелінійної регресії	Отримує довірчий інтервал нелінійної регресії для інтервального оцінювання вибіркового середнього розміру ПС для раннього прогнозування розміру програмних систем
Користувач	Отримати інтервал передбачення нелінійної регресії	Отримує інтервал передбачення нелінійної регресії для інтервального оцінювання вибіркового середнього розміру ПС для раннього прогнозування розміру програмних систем під управлінням платформи WordPress

Діаграма варіантів використання ПС прогнозування розміру представлена на рисунку 3.1.

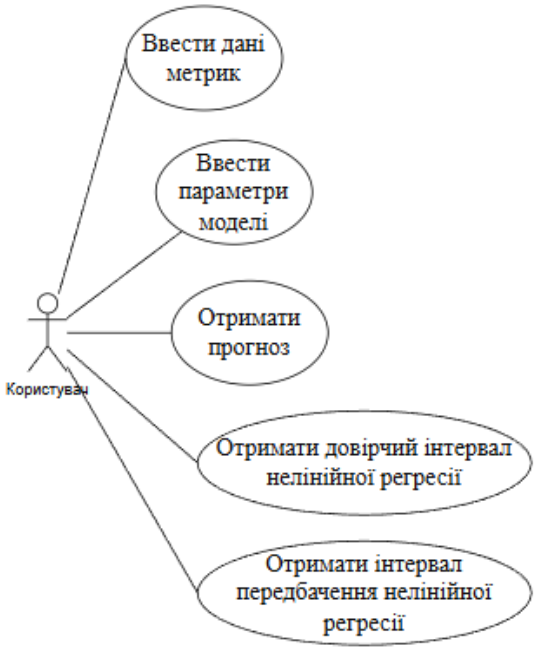

Рисунок 3.1 – Діаграма варіантів використання ПС прогнозування розміру

Таким чином, у роботі розроблена діаграма варіантів використання, яка визначає основні функції програмної системи та описує взаємодію користувачів із її ключовими можливостями.

3.2.3 Специфікації варіантів використання ПС прогнозування розміру

Для деталізації варіантів використання, необхідно розробити їх специфікації [11].

Прецедент: «Ввести дані метрик»

Актор: Користувач.

Короткий опис: Варіант використання відповідає за введення даних метрик.

Базовий потік подій: Введення даних метрик

Альтернативні потоки подій: Не визначена.

Передумова: Не визначена.

Післяумова: Не визначена.

Прецедент: «Ввести параметри моделі»

Актор: Користувач.

Короткий опис: Варіант використання відповідає за введення параметрів моделі.

Базовий потік подій: Введення параметрів моделі

Альтернативні потоки подій: Не визначена.

Передумова: Не визначена.

Післяумова: Не визначена.

Прецедент: «Отримати прогноз»

Актор: Користувач.

Короткий опис: Варіант використання відповідає за виведення даних раннього прогнозування розміру програмної системи.

Базовий потік подій: Виведення даних метрик

Альтернативні потоки подій: Не визначена.

Передумова: Не визначена.

Післяумова: Не визначена.

Прецедент: «Отримати довірчий інтервал нелінійної регресії»

Актор: Користувач.

Короткий опис: Варіант використання відповідає за виведення даних довірчого інтервалу нелінійної регресії.

Базовий потік подій: Виведення даних довірчого інтервалу нелінійної регресії.

Альтернативні потоки подій: Не визначена.

Передумова: Не визначена.

Післяумова: Не визначена.

Прецедент: «Отримати інтервал передбачення нелінійної регресії»

Актор: Користувач.

Короткий опис: Варіант використання відповідає за виведення даних інтервалу передбачення нелінійної регресії.

Базовий потік подій: Виведення даних інтервалу передбачення нелінійної регресії.

Альтернативні потоки подій: Не визначена.

Передумова: Не визначена.

Післяумова: Не визначена.

3.2.4 Сценарії основних варіантів використання ПС прогнозування розміру

Сценарії варіантів використання (Use Cases) - це текстові або діаграмні описи взаємодії користувача (актора) із системою для досягнення конкретної мети, що використовуються для проєктування, документування та тестування програмного забезпечення. Вони деталізують, «хто» і «що» може зробити із системою [11]. Для нашого проєкту було прийнято рішення розробити сценарії варіантів використання у вигляді діаграми діяльності яка зображена на рисунку 3.2.

Рисунок 3.2 – Діаграма діяльності ПС прогнозування розміру

Таким чином, у роботі розроблена діаграма діяльності, яка відображає логіку виконання процесів у системі та послідовність дій, що забезпечують реалізацію основних функціональних сценаріїв.

3.3 Рішення з інформаційного забезпечення для програмної системи прогнозування розміру

3.3.1 Концептуальна модель даних ПС прогнозування розміру

Концептуальне проєктування - це створення уявлення (схеми, моделі) даних, що включає визначення сутностей, атрибутів і зв'язків між ними, але не залежить від моделі даних (ієрархічної, мережевої, реляційної і т. д.) і фізичної реалізації [11].

Концептуальна модель даних ПС прогнозування розміру представлена на рисунку 3.3.

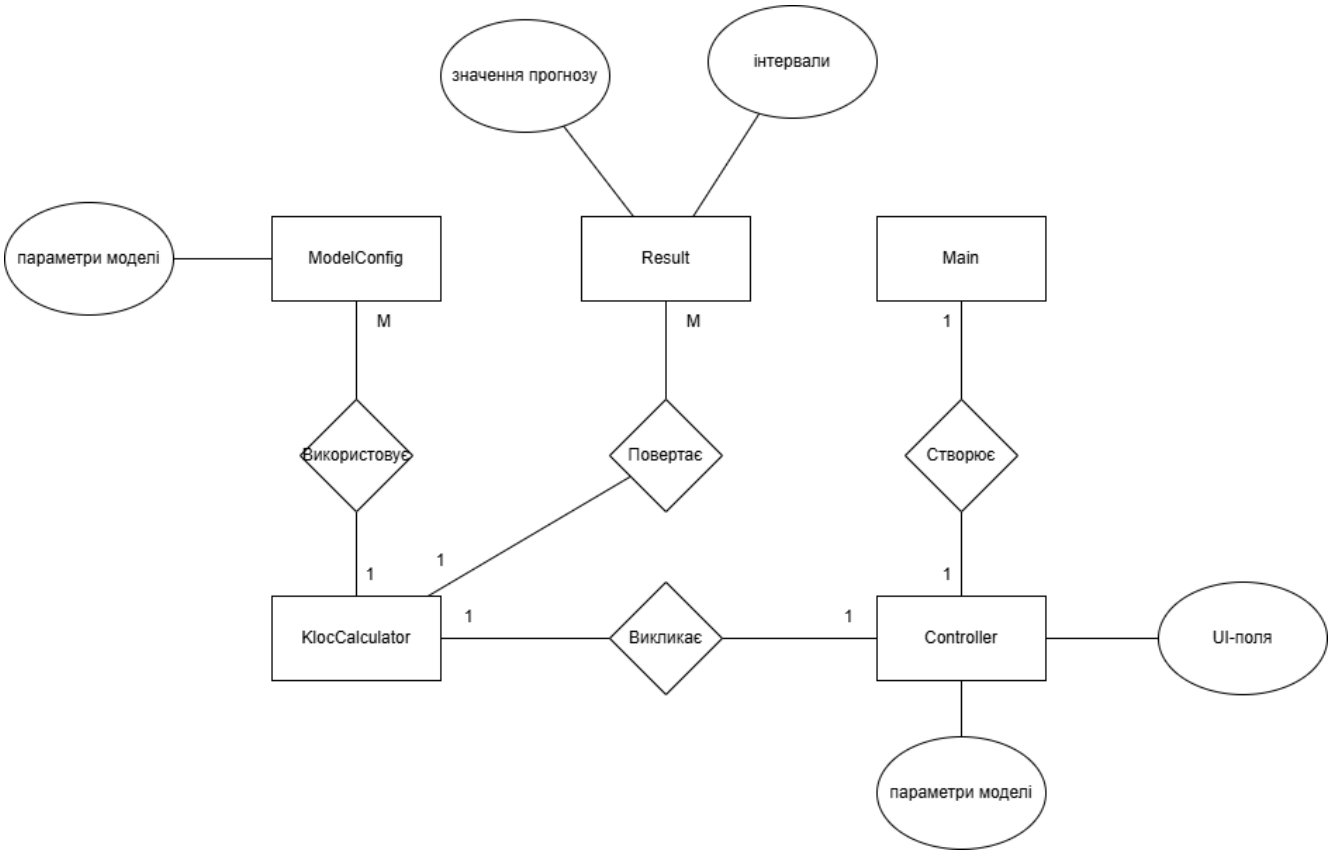

Рисунок 3.3 – Концептуальна модель даних ПС прогнозування розміру

Як бачимо, на діаграмі представлені основні сутності програмної системи, їх властивості та взаємозв'язки між ними.

3.3.2 Логічна модель даних ПС прогнозування розміру

Логічна модель даних - допомагає визначити структуру та взаємозв'язки даних у системі без прив'язки до конкретних технологій чи фізичної реалізації. Вона включає сутності, атрибути та зв'язки між ними, що дозволяє зрозуміти, як дані повинні бути організовані та керовані [11]. Логічну модель даних ПС прогнозування розміру зображено на рисунку 3.4.

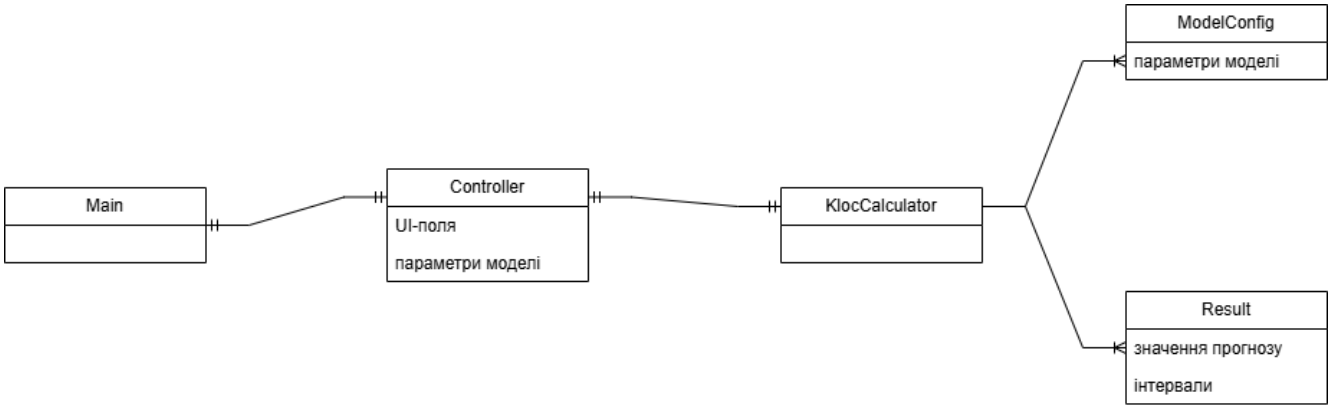

Рисунок 3.4 – Логічна модель даних ПС прогнозування розміру

Як бачимо, на діаграмі логічної моделі представлені основні програмні сутності системи, їх атрибути та взаємозв’язки, які забезпечують реалізацію процесу прогнозування розміру програмної системи.

3.4 Рішення з програмного забезпечення програмної системи прогнозування розміру

3.4.1 Вибір мови програмування для ПС прогнозування розміру

Вибір мови програмування є важливим етапом при створенні програмного забезпечення. Обрана мова має забезпечувати зручність написання та читання

коду, підтримку необхідних математичних операцій, а також надавати можливість створення сучасного й зрозумілого інтерфейсу користувача.

Для вирішення поставленої задачі доцільно обрати мову, що підтримує роботу зі складними математичними обчисленнями та має розвинуту екосистему. За цих умов повністю підходить мова програмування Java.

Java - це об’єктно-орієнтована, строго типізована мова програмування, що забезпечує високу продуктивність, переносимість між різними операційними системами та велику кількість сторонніх бібліотек для математичних і статистичних обчислень. У поєднанні з бібліотеками для наукових обчислень, такими як *Apache Commons Math*, Java стає потужним інструментом для побудови математичних моделей і виконання регресійного аналізу.

Серед переваг Java - безкоштовність, кросплатформеність, багаторічна стабільність, наявність великої кількості готових бібліотек, а також можливість створення графічного інтерфейсу будь-якої складності.

Для реалізації графічного інтерфейсу користувача та взаємодії з математичними алгоритмами буде використано JavaFX разом з FXML, які є сучасними інструментами для побудови інтерфейсів у Java. JavaFX забезпечує гнучке і зручне створення форм, кнопок, полів введення і дозволяє логічно відокремити програмний код від графічного оформлення.

Таким чином, мова Java разом із бібліотеками JavaFX, FXML і Apache Commons Math є оптимальним вибором для розробки програмного забезпечення, призначеного для раннього прогнозування розміру програмних систем під управлінням платформи WordPress.

3.4.2 Модель класів ПС прогнозування розміру

Діаграму класів використовують для зображення статичної структури проєктованої системи. Діаграма класів оперує поняттями класу, інтерфейсу, об'єкту, відношення та пакету.

З допомогою діаграми класів можна представляти концептуальний рівень

проєктування системи (поняття предметної галузі), рівень специфікацій (операції класу, які видимі ззовні), рівень реалізації (реалізацію класів) [11].

Діаграма класів ПС прогнозування розміру зображена на рисунку 3.5.

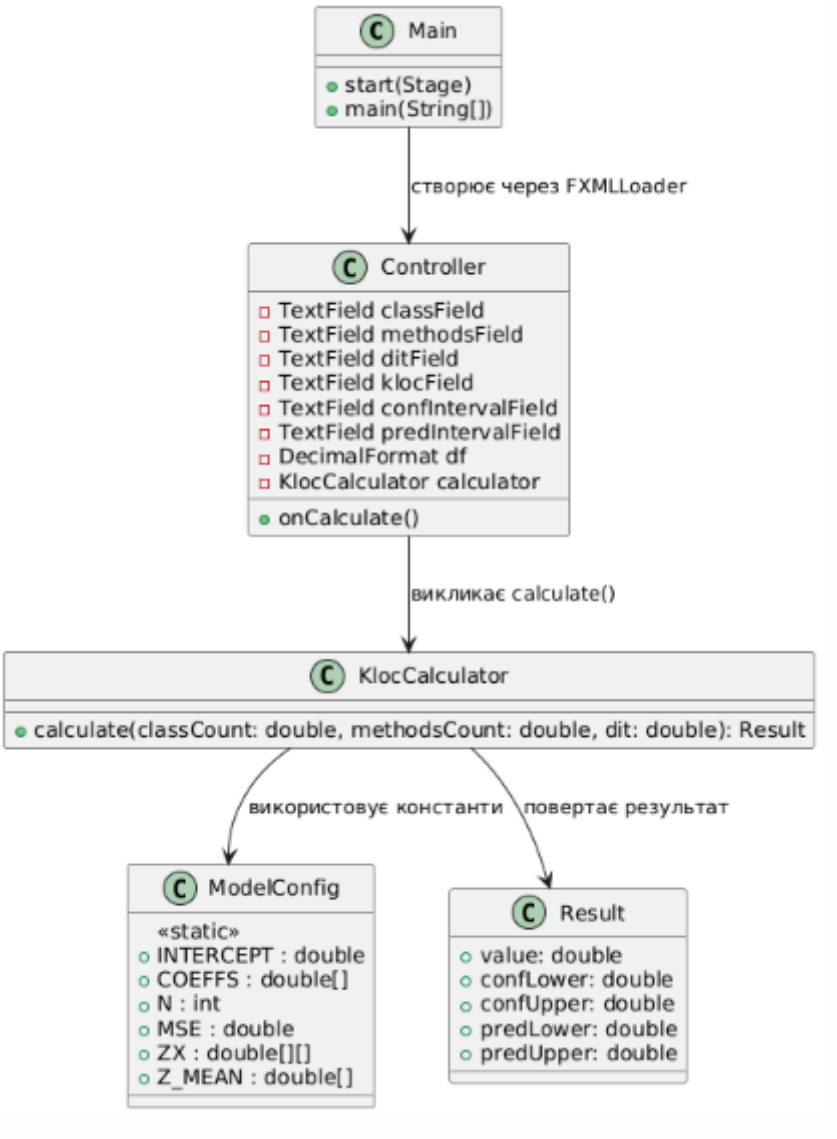

Рисунок 3.5 – Модель класів ПС прогнозування розміру

На діаграмі представлені наступні класи:

Клас Main - відповідає за запуск JavaFX-додатку та початкову ініціалізацію інтерфейсу.

Клас Controller - є UI-логікою, відповідає за зчитування даних з полів, виклик KlocCalculator та виведення результатів.

Клас ModelConfig - Зберігає всі константи моделі (коефіцієнти, матриці, середні значення).

Клас KlocCalculator - Виконує всі математичні розрахунки, включно з: основною формулою KLOC, обчисленням, формуванням інтервалів.

Клас Result - Це контейнер результатів обчислення моделі прогнозування, який використовується для передачі всіх вихідних значень між обчислювальною логікою та UI.

Таким чином, у роботі розроблена діаграма класів, яка представляє статичну структуру моделі системи.

3.4.3 Моделі взаємодії ПС прогнозування розміру

Діаграма взаємодії - це модель у мові UML, що візуалізує взаємодію між об'єктами в межах одного сценарію, показуючи послідовність повідомлень, якими вони обмінюються. Вона складається з ліній життя (вертикальні пунктирні лінії) для кожного об'єкта та стрілок, що позначають надсилання повідомлень, і допомагає зрозуміти динамічну поведінку системи [11].

Діаграму взаємодії ПС прогнозування розміру зображено на рисунку 3.6

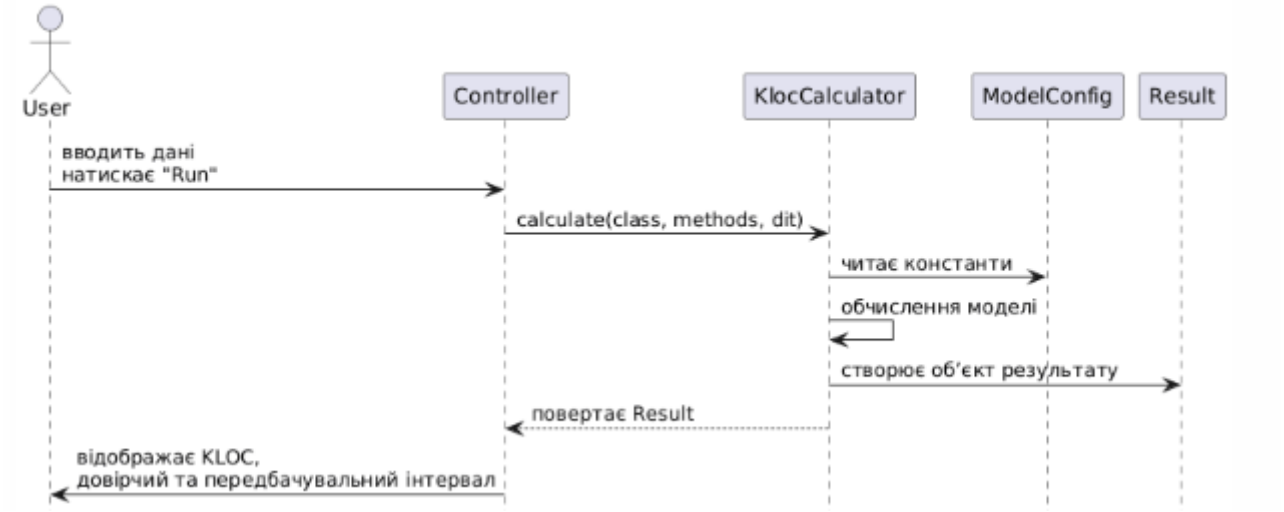

Рисунок 3.6 – Діаграма взаємодії ПС прогнозування розміру

Таким чином, у роботі розроблена діаграма взаємодії, яка відображає динамічний аспект роботи системи, демонструючи послідовність обміну повідомленнями між об’єктами під час виконання ключових сценаріїв.

3.4.4 Тестування ПС прогнозування розміру

Мета тестування - оцінити відповідність програмного забезпечення вимогам, які були сформульовані при постановці завдання, виявити помилки проєктування. Якщо достатній рівень відповідності програмного забезпечення був досягнутий, відсутні грубі помилки, то програмне забезпечення приймається до експлуатації.

Тестування будемо здійснювати з використанням Use Case. Таке тестування дає змогу виявити додаткові помилки у додатку, які складно знайти при тестуванні індивідуальних модулів, частин програми окремо одна від одної.

Тест 1 - Тестування можливості вивести на екран значення раннього прогнозу розміру програмних систем, довірчий інтервал та інтервал прогнозування для нелінійної моделі:

- Ввели дані для розрахунку раннього прогнозу розміру ПС.
- Натиснули на кнопку «Run».
- Розмір ПС, довірчий інтервал та інтервал прогнозування відобразилися на екрані.

Результат відповідає очікуваному.

3.4.5 Випробування ПС прогнозування розміру

Проведене функціональне випробування програмної системи методом чорної скриньки за даними з таблиці 2.1.

Протокол випробування:

Виконана перевірка на відповідність технічному завданню:

- перевірка функції вводу метрик ПС.

- перевірка функції вводу даних моделі.
- перевірка функції отримання розрахунку раннього прогнозу розміру ПС.
- перевірка функції отримання довірчого інтервалу.
- перевірка функції отримання інтервалу передбачення.

Графічне вікно програми з інтерфейсними компонентами зображено на рис. 3.7.

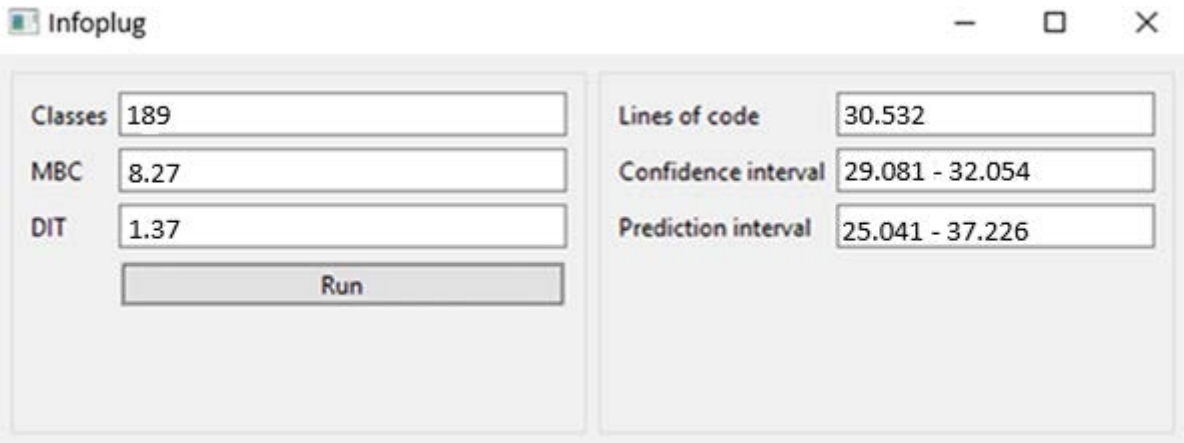

Рисунок 3.7 – Графічне вікно програмної системи з інтерфейсними компонентами

Таким чином, отримали значення раннього прогнозу розміру ПС, довірчий інтервал та інтервал прогнозування для нелінійної моделі.

Результати випробування показали, що розроблена програмна система повністю відповідає вимогам, що визначені в технічному завданні.

3.5 Висновки за розділом 3

У даному розділі було обґрунтовано вибір засобів моделювання для раннього прогнозування розміру програмних систем під управлінням платформи WordPress та мови програмування також були розроблені ескізний, технічний та робочій проєкти.

Загальносистемні рішення для програмної системи раннього прогнозування розміру включають розробку моделі варіантів використання з наведених у постановці задачі вимог до функцій програми, діаграму діяльності, зовнішню специфікацію програмної системи.

Рішення з інформаційного забезпечення програмної системи раннього прогнозування розміру включають розробку концептуальної моделі даних та логічної моделі даних програмної системи.

Рішення з програмного забезпечення програмної системи раннього прогнозування розміру включають вибір мови програмування для програмної системи, розробку моделі класів та моделі взаємодії програмної системи, розробку програму для раннього прогнозування розміру програмних систем під управлінням платформи WordPress та програмної документації, здійснення тестування та випробування створеної програмної системи, які показали, що створена програмна система повністю відповідає поставленим вимогам.

4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД РОЗРОБКИ І ВПРОВАДЖЕННЯ ПРОГРАМНОЇ СИСТЕМИ ДЛЯ РАННЬОГО ПРОГНОЗУВАННЯ РОЗМІРУ ПС ПІД УПРАВЛІННЯМ ПЛАТФОРМИ WORDPRESS

Дана кваліфікаційна робота присвячена створенню нелінійної регресійної моделі для раннього прогнозування розміру програмних систем під управлінням платформи WordPress. Також розглядається програмне забезпечення, необхідне для реалізації цієї моделі.

Економічна доцільність розробки цього програмного забезпечення може бути обґрунтована через розрахунок собівартості та очікуваного прибутку від його продажу. Головним критерієм, що дозволяє оцінити ефективність витрат на створення програмного продукту, є річний економічний ефект.

Обґрунтування доцільності розробки можна здійснити, оцінюючи собівартість програмного продукту та прибуток, який він принесе після виходу на ринок. Основним показником, що визначає економічну ефективність таких витрат, є річний економічний ефект.

Досягнення ефективного результату передбачає оптимізацію витрат: мінімум витрачених ресурсів на вході та максимальний результат на виході. Оцінка ефективності дозволяє визначити, наскільки обґрунтовані інвестиції та як оптимально використовуються ресурси.

Одним з ключових показників економічної ефективності програмного продукту є ефективність витрат, що визначається через строк окупності програмного забезпечення, який розраховується на етапі розробки та впровадження.

У цьому розділі буде розглянута собівартість розробки програмного продукту та проведена оцінка окупності проєкту, зокрема через можливість скорочення кількості працівників.

4.1 Розрахунок витрат на створення та впровадження програмної системи

Витрати на розробку програмного забезпечення складаються з витрат на заробітну плату розробника, на амортизацію та експлуатацію ПК, на якій виконується розробка, на засоби розробки та витрати на матеріали і комплектуючі.

Розробка ПЗ виконується завдяки спеціалісту з місячним окладом 23 200 грн. Плюс додаткова заробітна плата становить 10% від основної. Виходячи з цього зарплата спеціаліста становить 25520 грн./міс, а вартість сучасного комп`ютера з периферією складає 19000 грн. (середня вартість комп`ютера на базі процесору Intel Core i5 на 2025 рік). Вартість кіловат-годин електроенергії дорівнює 4.32 грн. В таблиці 4.1 зазначені витрати на допоміжні матеріали.

Таблиця 4.1 - Витрати на допоміжні матеріали

Пункт витрат	Сума, грн
Допоміжне програмне забезпечення	1500
Папір	200
Непередбачені витрати	400
Матеріали і комплектуючі вироби	1000

Вартість розробки системи розраховується за формулою:

$$Спр = (Ззп + Зсз + Взг + Зе) * Т + Зм, \quad (4.1)$$

де Т - тривалість розробки, міс.;

Ззп - основна і додаткова заробітна плата, грн.;

Зсз - відрахування на соціальні заходи (38% від основної і додаткової заробітної плати), грн.;

Взг - загальногосподарські витрати (10% від заробітної плати), грн.;

Зе - витрати на електроенергію;

Зм - витрати на основні і допоміжні матеріали.

Розрахуємо Зе при споживанні потужності 0,4 Вт, тривалості роботи на місяць рівної 19 ∗ 10 = 190 годин і вартості кіловат-години електроенергії 4,32 грн. до 100кВт включно та 5,5 після 100кВт отримаємо:

$$Зе = (100 * 4,32 + 90 * 5,5) * 0,4 = 370,8 \text{ грн.}$$

Представимо всі поточні витрати на розробку програмного забезпечення в таблиці 4.2

Таблиця 4.2 - Витрати на розробку програмного забезпечення

Витрати	Одиниця виміру	Кількість
Тривалість розробки	міс	3
Заробітна плата(осн + додаткова)	грн	25520
Відрахування на соц заходи	грн	9697,6
Загальногосподарські витрати	грн	1500
Витрати на допоміжні матеріали	грн	3100
Витрати на електроєнергію	грн	370,8

Виходячи з наявних показників вартості можна розрахувати вартість розробки ПЗ:

$$Спр = (25520 + 9697,6 + 1500 + 370,8) * 3 + 3100 = 114\,313,2 \text{ грн.}$$

Амортизаційні відрахування на устаткування складають 60% балансової вартості в рік:

$$Аоб = 19000 * 0,6 = 11400 \text{ грн.}$$

У масштабах підприємства річні витрати на основні і допоміжні матеріали визначаються в розмірі 5% вартості основного устаткування:

$$Вм = 23200 * 0,05 = 1160 \text{ грн.}$$

Річний обсяг робіт ПК у годинах визначається в такий спосіб:

$$Фм = 253,3 * Тз,$$

де Тз - це середнє місячне завантаження устаткування (близько 7 годин), 253,3 - середня кількість робочих днів у році.

$$Фм = 253,3 * 7 = 1773,1 \text{ год.}$$

Витрати на електроенергію Зе при 1773,1 годинах роботи устаткування в рік складуть:

$$Зе = 1773,1 * 0,4 * 4,32 = 3063,9 \text{ грн.}$$

Експлуатаційні витрати для ПК за рік складуть:

$$Ззр = 11400 + 1160 + 3063,9 = 15623,9 \text{ грн.}$$

У перший рік витрати на створення й експлуатацію програмного забезпечення складуть:

$$Зсе = 114313,2 + 15623,9 = 129037,1 \text{ грн.}$$

4.2 Розрахунок економічної ефективності розробки та впровадження програмної системи

Визначити пряму економічну ефективність можна, ґрунтуючись на тому, що впровадження програмного забезпечення вивільняє одного працівника (за експертною оцінкою фахівців підприємства).

Зарплата одного працівника в рік складає:

$$23200 * 12 * 1 = 278400 \text{ грн.}$$

Річний економічний ефект розраховується по формулі:

$$Эрік = \Delta Cn - En * k, \quad (4.2)$$

де ΔCn - вивільнені кошти після впровадження системи (278400 грн.) мінус експлуатаційні витрати (15623,9 грн.);

ΔEn - коефіцієнт ефективності(дорівнює коефіцієнту амортизації (0,6);

k - одноразові витрати на впровадження продукту (129037,1).

$$Эрік = 278400 - 15623,9 - 129037,1 * 0,6 = 185353,84 \text{ грн.}$$

Строк окупності системи розраховується по формулі:

$$T = \frac{k}{\Delta C} = \frac{129037,1}{278400 - 15623,9} \approx 0,49$$

Отже, строк окупності програмного забезпечення складає приблизно 5-6 місяців.

4.3 Висновки за розділом 4

В розділі були проведені розрахунки по створенню і впровадженню програмної системи для раннього прогнозування розміру ПС під управлінням платформи WordPress, після чого була розрахована економічна ефективність та строк окупності програмної системи. Як висновок з розрахунків, після впровадження програмної системи строк окупності буде складати 5-6 місяців протягом першого року. Отже, можна зробити висновок, що розробка цієї програмної системи є економічно виправданою та прибутковою.

5 ОХОРОНА ПРАЦІ

Охорона праці розглядається як цілісна система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних та лікувально-профілактичних заходів і засобів, основною метою яких є збереження життя, здоров'я та працездатності людини під час трудової діяльності. З правової точки зору вона являє собою сукупність норм, що регулюють відносини у сфері забезпечення безпечних і здорових умов праці. У ширшому розумінні охорону праці можна визначити як систему гарантування безпеки праці, що ґрунтується на законодавчих положеннях та комплексі соціально-економічних, технічних, організаційних і гігієнічних заходів.

Стан умов праці на робочому місці, безпека технологічних процесів, машин, механізмів і обладнання, а також рівень забезпечення працівників засобами колективного та індивідуального захисту і належні санітарно-побутові умови мають відповідати чинним нормативно-правовим актам у сфері охорони праці.

Обов'язком власника або уповноваженого ним органу є впровадження сучасних засобів безпеки праці, спрямованих на попередження виробничого травматизму, а також створення належних санітарно-гігієнічних умов, що мінімізують ризик виникнення професійних захворювань. Крім того, на роботодавця покладається відповідальність за регулярне проведення навчання та інструктажів з питань охорони праці й пожежної безпеки.

Охорона праці займає провідне місце серед інститутів трудового права та має важливе практичне значення. Порушення встановлених вимог у цій сфері становить серйозну загрозу для життя і здоров'я працівників, а особи, відповідальні за організацію праці, несуть за такі порушення сувору, зокрема й кримінальну, відповідальність [12].

5.1 Аналіз шкідливих та небезпечних факторів у офісі комп’ютерної фірми

Небезпечним виробничим фактором вважають такий чинник, дія якого за певних умов може призвести до отримання травми або до раптового суттєвого погіршення стану здоров’я працівника. У разі, коли вплив виробничого фактора зумовлює розвиток захворювання чи зниження працездатності, його відносять до шкідливих. Водночас за високої інтенсивності або тривалого впливу шкідливий фактор може трансформуватися в небезпечний [13].

У процесі офісної діяльності на працівника впливають дві основні групи шкідливих факторів:

Фізичні, що виникають унаслідок дії навколишнього середовища, зокрема електромагнітні випромінювання, рівень освітленості робочої зони, температура повітря та шум;

Психофізіологічні, пов’язані з перевантаженням організму, а саме зоровим напруженням, статичними навантаженнями та нервово-психічною напругою.

Освітлення являє собою процес створення, розподілу та раціонального використання світлової енергії з метою забезпечення комфортних і безпечних умов праці. Світлове середовище не лише дає змогу сприймати об’єкти праці, а й істотно впливає на психоемоційний стан людини та перебіг фізіологічних процесів. Правильно організоване освітлення у виробничих і офісних приміщеннях сприяє підвищенню продуктивності та безпеки праці, зменшенню втомлюваності й рівня травматизму, а також підтриманню високої працездатності.

Згідно з санітарно-гігієнічними вимогами, освітлення повинно бути достатнім за інтенсивністю, рівномірно розподіленим, не викликати засліплення та не створювати відблисків на робочих поверхнях. За спектральними характеристиками світло має бути максимально наближеним до природного сонячного. Нормативними документами рекомендовано переважно використовувати природне освітлення, оскільки воно найкраще сприймається органами зору. Штучне освітлення у виробничих та адміністративно-громадських

приміщеннях зазвичай організовується за системою загального рівномірного освітлення, при цьому допускається застосування комбінованої системи. Освітленість робочої поверхні столу в зоні розміщення документів повинна становити 300–500 лк, а як джерела світла доцільно використовувати переважно люмінесцентні лампи типу ЛБ.

Одним із найбільш несприятливих чинників, що негативно впливають на уважність і працездатність працівників, підвищують ризик травматизму та розвитку захворювань, є шум. Підвищений рівень шуму та вібрації чинить шкідливий вплив не лише на органи слуху, а й на нервову, серцево-судинну та інші системи організму, спричиняючи погіршення слуху, розвиток гіпертонії, психоемоційні розлади тощо.

Допустимі рівні звукового навантаження для працівників різних професій регламентуються гігієнічними нормативами (СН 3222-85, ГОСТ 12.1.003-85, ГР 2411-81) [14]. Відповідні нормативні значення для осіб, які працюють з персональними комп’ютерами, наведені в таблиці 5.1.

Таблиця 5.1 - Допустимі рівні шуму

Вид трудової діяльності	Рівні звукового тиску в дБ октавних смугах із середньо геометричними частотами, ГЦ									
	1,5	3	25	50	00	000	000	000	000	Рівні звуку, еквівалентні рівні звуку, дБа/дБАекв
Програмісти ПК	6	1	1	4	9	5	2	0	8	50
Оператори в залах обробки інформації на ПК ті оператори комп’ютерного набору	6	3	4	8	3	0	7	5	4	65
В приміщеннях для розташування шумних агрегатів ПК	03	1	3	7	3	0	8	6	4	75

Відповідно до чинних нормативів, під час реалізації заходів зі зниження рівня шуму усувається не вся віброакустична активність обладнання, а лише та її частина, що перевищує допустимі граничні значення та чинить негативний вплив на здоров’я людини. Основним напрямом боротьби з виробничим шумом є удосконалення технологічних процесів і конструкцій обладнання, насамперед за рахунок зменшення шуму від найбільш потужних його джерел. Водночас доцільно здійснювати комплексне знешумлення всього обладнання, що цього потребує, оскільки зниження шуму лише окремих установок за наявності значної кількості інших шумових джерел є малоефективним.

Фізичний стан і працездатність людини значною мірою визначаються тепловим станом організму, на який впливають параметри виробничого середовища, зокрема температура та вологість повітря, швидкість його руху, атмосферний тиск і вміст кисню. Так, зменшення концентрації кисню в повітрі до рівня близько 12 % призводить до утруднення дихання, підвищення його частоти та напруження дихальної системи, що може бути витримано організмом не більше ніж протягом пів години.

Перегрівання організму виникає за умов надмірного теплового впливу, зумовленого конвекцією та випромінюванням від нагрітих поверхонь. У таких ситуаціях активізуються захисно-пристосувальні механізми: посилюється робота серцево-судинної й дихальної систем, збільшується потовиділення, обсяг якого може досягати до 5 літрів за робочу зміну. Разом із потом організм втрачає значну кількість мінеральних речовин і вітамінів.

Вологість повітря є важливим чинником терморегуляції організму. Підвищена відносна вологість у виробничих приміщеннях ускладнює процеси теплообміну та зменшує ефективність тепловіддачі. Оптимальним з фізіологічної точки зору вважається рівень відносної вологості в межах 40–60 %.

Сукупність перелічених параметрів навколишнього середовища характеризує мікроклімат приміщення. Для приміщень, у яких використовуються персональні комп’ютери, встановлено відповідні нормативи мікроклімату,

наведені в таблиці 5.2.

Оскільки функціонування комп’ютерної техніки та периферійних пристроїв пов’язане з використанням електричної енергії, існує потенційна загроза ураження користувачів електричним струмом. З метою запобігання нещасним випадкам необхідно неухильно дотримуватися вимог і правил електробезпеки.

Таблиця 5.2 - Норми мікроклімату для приміщень з ПК

Пора року	Категорія робіт	Температура повітря, С, не більше	Відносна вологість повітря, %	Швидкість руху повітря, м/с
Холодна	Легка - 1а	22-24	40-60	0,1
	Легка - 1б	21-23	40-60	0,1
Тепла	Легка - 1а	23-25	40-60	0,1
	Легка - 1б	22-24	40-60	0,2

Небезпека ураження електричним струмом, на відміну від багатьох інших шкідливих чинників, посилюється тим, що людина без спеціальних технічних засобів не може на відстані визначити наявність напруги. Крім того, дія електричного струму має миттєвий характер, і реальна загроза стає очевидною вже після факту ураження. Вплив струму на організм людини призводить до різноманітних порушень, проявляючись як у вигляді локальних ушкоджень тканин і органів, так і у формі загального ураження всього організму.

Подразнювальну дію на людину чинить змінний електричний струм промислової частоти силою приблизно 0,6–1,5 мА, а також постійний струм у межах 5–7 мА. Такі значення, як правило, не становлять значної небезпеки, оскільки за цих умов людина здатна самостійно припинити контакт зі струмоведучими частинами. Для змінного струму промислової частоти межі так званого «невідпускаючого» струму становлять 6–20 мА. Постійний струм, на відміну від змінного, не викликає ефекту мимовільного утримання, однак супроводжується сильними больовими відчуттями; при цьому сила такого струму може досягати 15–80 мА і більше.

5.2 Розрахунок системи пожежогасіння у офісі комп’ютерної фірми

Розрахунок системи пожежогасіння - це процес визначення параметрів, за якими система повинна подавати вогнегасну речовину (воду, піну, газ чи порошок) для ефективного придушення пожежі. Такий розрахунок дає змогу визначити необхідний тиск, витрату, діаметр трубопроводів, тип і кількість пожежних розбризкувачів або модулів, а також обсяг резервуарів та продуктивність насосів [15].

Основна мета розрахунку - забезпечити надійне й швидке гасіння пожежі на ранньому етапі, мінімізувати збитки та гарантувати безпеку людей і обладнання. Розрахунок потрібен для відповідності нормативним вимогам, правильного вибору обладнання та забезпечення працездатності системи в реальних умовах [16]. Групи приміщень наведені в таблиці 5.3.

Таблиця 5.3 - Групи приміщень

Група	Приміщення	Пожежне навантаження
1	Приміщення книгосховищ, ПК, магазинів, будівель управлінь, готелів, лікарень	до 200 Мдж/м^2
2	Приміщення з використанням рідин, які легко спалахують (ЛСР) і горять (ГР); деревообробні, швейні та інші приміщення; підприємства з обслуговування автомобілів	200...2000 Мдж/м^2
3	Приміщення гумотехнічного виробництва	200...2000 Мдж/м^2
4	Приміщення для виробництва, обробки, переробки різних матеріалів з використанням ЛСР і ГР	>2000 Мдж/м^2
5	Склади негорючих матеріалів в упаковці, що горить	>2000 Мдж/м^2
6	Склади твердих матеріалів, що горять	>2000 Мдж/м^2
7	Склади лаків, фарб, ЛСР, ГР, пластмас та ін.	>2000 Мдж/м^2

Група приміщення: 1 Тип зрошувача: вода

Розміри приміщення: довжина a = 8 м, ширина b = 7 м

1. Знайдемо групу приміщення (табл. 5.3) згідно з пожежним навантаженням, які забезпечуються автоматичними установками пожежогасіння (ДВН В.2.5-13-98).

За таблицею 5.3 приміщення офісу комп’ютерної фірми належить до 1

групи.

2. Параметри для розрахунку спринклерної установки залежно від групи приміщення (1) є наступними:

Інтенсивність зрошування водою L = 0,08 л/(см2)

Площа, яка захищається одним зрошувачем Sзр = 12 м2

Тривалість роботи установки водного пожежогасіння T = 30 хвилин

Відстань між зрошувачами D = 4m

Знаходимо площу приміщення:

$$S_{прим} = a * b = 8 * 7 = 56 \text{ м}^2$$

Знаходимо необхідну кількість зрошувачів:

$$N = S_{прим}/S_{зр} = 56 / 12 = 4,7$$

Розміщуємо зрошувачі на плані приміщення.

Знаходимо необхідну інтенсивність води в трубопроводі:

$$L_{тр} = L * S_{прим} = 0,08 * 56 = 4,48 \text{ л/с}$$

Знаходимо інтенсивність води крізь один спринклер:

$$L_{др} = L_{тр} / N = 4,48 / 4,7 = 0,95 \text{ л/с}$$

Знаходимо необхідний об’єм води:

$$Q_н = L_{тр} * T = 4,48 * 1800 \text{ с} = 8064 \text{ л}$$

5.3 Розробка заходів щодо зменшення впливу шкідливих та небезпечних факторів

Сучасна ергономіка виходить з того, що універсального положення тіла для роботи протягом усього дня не існує. Тому робоче місце має бути регульованим і пристосованим щонайменше до кількох робочих поз, з урахуванням характеру виконуваних завдань та індивідуальних особливостей працівника. Найбільш доцільними вважаються вертикальне або злегка нахилене положення тіла з можливістю регулювання крісла та монітора.

Розміщення робочих місць з персональними комп’ютерами повинно забезпечувати нормативні відстані між моніторами: не менше 2 м у напрямку «тил–екран» та не менше 1,2 м між їх бічними поверхнями. Робочий стіл має забезпечувати достатній простір для ніг відповідно до ергономічних вимог.

Положення монітора повинно відповідати напрямку погляду користувача, оскільки неправильна висота або кут нахилу екрана призводять до сутулості та перевтоми. Оптимальна відстань від очей до дисплея становить 40–70 см. Для зменшення навантаження на зір рекомендується використовувати достатньо великий шрифт, уникати мерехтіння екрана та робити регулярні перерви для відпочинку очей. Частота оновлення зображення має бути не нижчою за 72 Гц.

Монітор не слід розташовувати навпроти або під кутом до вікна, щоб уникнути відблисків і надмірного освітлення. У разі розміщення робочого місця біля вікна екран повинен бути перпендикулярним до нього. Для зменшення впливу електромагнітного випромінювання доцільно використовувати монітори, що відповідають стандартам MPR II або TCO, або застосовувати захисні фільтри.

Крісло має підтримувати спину та регулюватися за висотою, забезпечуючи комфортне положення без тиску на стегна чи куприк. Клавіатуру необхідно розміщувати так, щоб не доводилося нахилятися вперед; при цьому руки повинні бути зігнуті в ліктях приблизно під прямим кутом, а зап’ястки залишатися прямими. Висота столу та крісла має узгоджуватися між собою, за потреби слід використовувати підставку для ніг.

Під час роботи за комп’ютером рекомендовано щогодини робити перерви та виконувати прості вправи для кистей і пальців. Дотримання наведених ергономічних рекомендацій сприяє зменшенню негативного впливу шкідливих факторів та підвищенню комфорту і безпеки праці.

6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

6.1 Вступ

Навколишнє (природне) середовище - це сукупність природних об’єктів і умов, у межах яких здійснюється діяльність людини та інших суб’єктів, що характеризуються здатністю до самопідтримки й саморегуляції без безпосереднього впливу людини.

Охорона навколишнього середовища (ОНС) розглядається як система законодавчих і нормативних актів, стандартів та інструкцій, які встановлюють вимоги щодо збереження природи під час будівництва, реконструкції та експлуатації промислових, громадських і житлових об’єктів. Реалізація ОНС передбачає комплекс технічних і організаційних заходів, спрямованих на захист атмосферного повітря, поверхневих і підземних вод, відновлення порушених земель, раціональне використання ґрунтів, охорону надр і тваринного світу.

Стрімкий розвиток науки, техніки та промисловості, характерний для сучасного етапу, супроводжується зростанням негативного впливу на довкілля. Недосконаліть технологічних процесів зумовлює утворення значних обсягів промислових відходів, які щороку у вигляді твердих, рідких і газоподібних речовин потрапляють у біосферу, завдаючи шкоди природним екосистемам.

Інтенсивні зміни середовища проживання порушують взаємозв’язок між людиною та природою, знижують адаптаційні можливості організму і негативно впливають на стан здоров’я населення. Тому оцінка здоров’я людини є неможливою без урахування екологічних змін. Важливу роль у цьому відіграє функціонування державної інформаційної системи «здоров’я населення — навколишнє середовище», основним завданням якої є збирання та аналіз даних про рівень забруднення довкілля і стан здоров’я населення.

6.2 Аналіз існуючої проблеми необхідності охорони навколишнього середовища

Охорона навколишнього природного середовища, раціональне використання природних ресурсів і забезпечення екологічної безпеки є необхідними передумовами сталого соціально-економічного розвитку України. З цією метою держава реалізує екологічну політику, спрямовану на збереження безпечного для життя довкілля, захист здоров’я населення від негативного впливу забруднення, досягнення збалансованої взаємодії суспільства і природи, а також на охорону, відтворення та ефективне використання природних ресурсів.

Закон України про охорону навколишнього природного середовища визначає правові, економічні та соціальні засади діяльності у цій сфері в інтересах як нинішнього, так і майбутніх поколінь. Основними завданнями екологічного законодавства є регулювання відносин у галузі використання й охорони природних ресурсів, забезпечення екологічної безпеки, попередження та усунення негативних наслідків господарської діяльності, збереження природних комплексів, ландшафтів, біорізноманіття та об’єктів, що мають історико-культурну цінність.

Охорона довкілля ґрунтується на низці базових принципів, серед яких пріоритет екологічної безпеки, обов’язкове дотримання встановлених нормативів і стандартів, гарантування безпечних умов життя для населення, запобіжний характер природоохоронних заходів та екологізація виробництва. Важливу роль відіграють збереження природної різноманітності, науково обґрунтоване поєднання екологічних, економічних і соціальних інтересів, обов’язковість державної екологічної експертизи та відкритість у прийнятті рішень.

Крім того, законодавством передбачено відповідальність за шкоду, заподіяну довкіллю, поєднання стимулюючих і контрольних заходів, урахування рівня антропогенного навантаження на території, розвиток міжнародного співробітництва та застосування економічних механізмів регулювання, зокрема екологічних податків і зборів відповідно до Податкового кодексу України [17].

6.3 Забруднення навколишнього середовища співробітниками офісу комп’ютерної фірми

У процесі функціонування офісу комп’ютерної фірми утворюються тверді побутові відходи, до яких належать папір, картон, пакування та інші типові офісні відходи. Проблема ТПВ є актуальною не лише для окремих установ, а й для міст і держави загалом, зокрема й для України, де її масштаби співвідносні зі світовими.

За умов підвищеної вологості та високої температури відходи зазнають анаеробного розкладання з утворенням шкідливих речовин. Несанкціоноване складування ТПВ призводить до забруднення ґрунтів, підземних і поверхневих вод, а також створює загрозу для здоров’я людей. Особливо небезпечними такі смітники є в теплу пору року за температури понад +25 °С, коли активізуються процеси гниття та поширення патогенних мікроорганізмів, що може сприяти виникненню інфекційних захворювань.

Найпоширенішим способом поводження з ТПВ наразі залишаються полігони. Водночас їх використання супроводжується низкою суттєвих недоліків: швидким переповненням, негативним впливом на довкілля (забруднення вод, утворення метану, пожежі, неприємні запахи), дефіцитом придатних земель поблизу міст і зростанням витрат на транспортування. Крім того, повністю відмовитися від полігонів неможливо, оскільки вони залишаються необхідними для захоронення непереробляваних відходів [18].

6.4 Розробка заходів щодо зменшення забруднення

Збір відходів є одним із найдорожчих етапів системи поводження з ТПВ, тому його раціональна організація дає змогу суттєво зменшити витрати. В Україні система збору відходів має залишатися економічно обґрунтованою та

стандартизованою, водночас потребуючи додаткового планування для вирішення нових проблем, зокрема збору відходів від комерційних об’єктів. У густонаселених районах доцільним є створення станцій тимчасового зберігання, які дозволяють оптимізувати транспортування, хоча такі об’єкти потребують суворого дотримання екологічних вимог.

Значна частина твердих побутових відходів може бути залучена до вторинної переробки. Скло переробляють шляхом подрібнення та переплавлення або використовують як наповнювач у будівельних матеріалах. Металеві банки підлягають переплавленню, причому повторне використання алюмінію є енергетично дуже вигідним. Паперові відходи застосовують для виготовлення картону, санітарно-гігієнічного паперу, теплоізоляційних матеріалів та в сільському господарстві, хоча в Україні такі технології розвинені обмежено. Переробка пластику залишається складною і затратною, а отримана вторинна сировина не завжди має високі споживчі властивості.

Компостування є природним методом переробки органічних відходів і широко використовується як у побуті, так і на спеціалізованих майданчиках. Отриманий компост може застосовуватися в міському та сільському господарстві. Водночас централізоване компостування змішаних ТПВ, що використовується на окремих підприємствах в Україні, дає продукт обмеженого застосування через потенційну небезпеку.

Отже, проблема побутових відходів залишається актуальною та потребує впровадження сучасних технологій утилізації й активної участі населення у збереженні довкілля [19].

ВИСНОВКИ

Кваліфікаційна робота охоплює повний цикл дослідження стосовно раннього прогнозування розміру програмних систем під управлінням платформи WordPress. Проведена робота забезпечила комплексний підхід до вирішення поставленої практичної задачі: від ґрунтовного аналізу існуючих підходів до побудови та перевірки власної моделі, а також створення програмної системи, що реалізує результати дослідження. Поставлені завдання були реалізовані у повному обсязі:

- виконаний аналіз існуючих методів та моделей раннього прогнозування розміру програмних систем під управлінням платформи WordPress;

- аргументована необхідність удосконалення математичної моделі для раннього прогнозування розміру програмних систем під управлінням платформи WordPress;

- зібрані дані для побудови математичної моделі, які нормалізовані, використовуючи перетворення у вигляді десяткового логарифму, та перевірені на викиди;

- побудована лінійна регресійна модель (лінійне рівняння регресії, довірчий інтервал та інтервал прогнозування) для даних, нормалізованих через перетворення у вигляді десяткового логарифму;

- побудована нелінійна регресійна модель (нелінійне рівняння регресії, довірчий інтервал та інтервал прогнозування) для раннього прогнозування розміру програмних систем під управлінням платформи WordPress;

- розроблена програмна система для раннього прогнозування розміру ПС під управлінням платформи WordPress.

На основі нормалізуючого перетворення у вигляді десяткового логарифму була підвищена достовірність раннього прогнозування розміру програмних систем під управлінням платформи Wordpress.

Технічне завдання представлене в додатку А. Код програмної системи представлений в додатку Б. Опис програмної системи представлений в додатку В. Інструкція користувача - в додатку Г. Програма і методика випробувань - у додатку Д.

Програмна система для раннього прогнозування розміру ПС під управлінням платформи WordPress, призначена для використання на персональних комп'ютерах, повністю пройшла тестування та випробування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Приходько, С. Б. Оцінювання розміру PHP-застосунків з відкритим кодом за нелінійними регресійними моделями з різними факторами / С. Б. Приходько, М. В. Ворона // *Збірник наукових праць НУК: Комп'ютерні науки та інформаційні технології*. - № 1. - Миколаїв: НУК імені адмірала Міакарова, 2021.

2. Проценко, В. С. Передобробка даних для побудови математичної моделі раннього прогнозування розміру програмних систем під управлінням платформи WordPress / В. С. Проценко, Л. М. Макарова, В. О. Каіров // *Матеріали VIII Всеукраїнської науково-практичної інтернет-конференції «Сучасні комп'ютерні системи та мережі в управлінні»* (24 листопада 2025 р.) / за ред. А. А. Григорової. - Херсон: ФОП Вишемирський В. С., 2025. - С. 172-173.

3. Огляд CMS Wordpress: що це, плюси та мінуси, приклади сайтів на Вордпресс. URL: https://www.interkassa.com/ua/blog/obzor-cms-wordpress-chto_eto-plyusy-i-minusy-primery-saytov-na-vordpress/

4. Електронний конспект лекцій з дисципліни "Емпіричні методи програмної інженерії". URL: https://ntuukpi.github.io/fiot/materials/EMPI_konspect.pdf

5. Приходько С.Б., Макарова Л.М., Приходько Н.В., Пухалевич А.В. Методичні вказівки та завдання до виконання лабораторних робіт з дисципліни "Емпіричні методи програмної інженерії". Миколаїв: НУК, 2023. 56 с.

6. Prykhodko, S. Detecting Outliers in Multivariate Non-Gaussian Data on the Basis of Normalizing Transformations / S. Prykhodko, N. Prykhodko, L. Makarova, K. Pugachenko // *Proceedings of the 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)*. - Kyiv, 2017. - P. 846-849.

7. Mardia K.V. Measures of multivariate skewness and kurtosis with applications. Biometrika. 57. 1970. P. 519–530. DOI: 10.1093/biomet/57.3.519

8. Prykhodko, S. Application of the Squared Mahalanobis Distance for Detecting Outliers in Multivariate Non-Gaussian Data / S. Prykhodko, N. Prykhodko, L.

Makarova, A. Pukhalevych // *Proceedings of TCSET*. - Lviv-Slavske, 2018. - P. 962-965.

9. Приходько, С. Б. Метод покращення нелінійних регресійних моделей на основі багатовимірних нормалізуючих перетворень / С. Б. Приходько, Н. В. Приходько // *Прикладні науково-технічні дослідження*: матеріали III міжнар. наук.-практ. конф. - Івано-Франківськ: Сімфоня Форте, 2019. - С. 20.

10. Prykhodko, N. V. Constructing the Non-Linear Regression Models on the Basis of Multivariate Normalizing Transformations / N. V. Prykhodko, S. B. Prykhodko // *Electronic Modeling*. - 2018. - Vol. 40, No. 6. - P. 99-108. - DOI: 10.15407/emodel.40.06.101.

11. Fowler, M. UML Distilled: A Brief Guide to the Standard Object Modeling Language. - 3rd ed. - Boston: Addison-Wesley, 2004. https://ptgmedia.pearsoncmg.com/images/9780321193681/samplepages/9780321193681_Sample.pdf (дата звернення: 06.11.2025)

12. Закон України «Про охорону праці» // *Відомості Верховної Ради України*. - 1992. - № 49. URL: https://zakon.rada.gov.ua/laws/show/2694-12#Text (дата звернення: 06.11.2025)

13. ДСТУ ISO 45001:2019. Системи управління охороною здоров’я та безпекою праці. URL: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=88004 (дата звернення: 06.11.2025)

14. ГОСТ 12.1.003-85 Гігієнічні нормативи. URL: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=48130 (дата звернення: 01.12.2025)

15. ISO 6182-1:2019. Fire protection — Automatic sprinkler systems. URL: https://online.budstandart.com/ru/catalog/doc-page.html?id_doc=86822 (дата звернення: 01.12.2025)

16. ДБН В.2.5-56:2014. Системи протипожежного захисту. URL: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=59526 (дата звернення: 01.12.2025)

17. Аналіз проблем забруднення поверхневих та підземних вод України та заходів щодо зменшення забруднення. URL: https://er.knutd.edu.ua/bitstream/123456789/11788/1/NRMSE2018_V2_P681-682.pdf (дата звернення: 15.12.2025).

18. Лекція з основ екології та видів забруднень (хімічні, фізичні, біологічні). URL: https://kfr.cnu.edu.ua/wp-content/uploads/sites/98/2020/11/1-%D0%BA%D1%83%D1%80%D1%81_%D0%A1%D0%B8%D0%BB%D0%B0%D0%B1%D1%83%D1%81-%D0%92%D0%B0%D0%BB%D0%B5%D0%BE%D0%BB%D0%BE%D0%B3%D1%96%D1%8F-2020.pdf (дата звернення: 15.12.2025).

19. Матеріали про технологічні, хімічні, біологічні та організаційні методи зменшення забруднення. URL: https://naurok.com.ua/metodi-borotbi-iz-zabrudnennyam-navkolishnogo-seredovischa-493702.html (дата звернення: 15.12.2025).

ДОДАТОК А - ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ ПРОГРАМНОЇ СИСТЕМИ РАННЬОГО ПРОГНОЗУВАННЯ РОЗМІРУ ПС ПІД УПРАВЛІННЯМ ПЛАТФОРМИ WORDPRESS «INFOPLUG»

Вступ

Назва розробки - Програмна система «INFOPLUG» раннього прогнозування розміру ПС під управлінням платформи WordPress.

Підстави для розробки

Підставою для розробки є завдання на кваліфікаційну роботу, видане кафедрою ПЗАС НУК ім. адмірала Макарова.

Призначення розробки:

Програма призначена для раннього прогнозування розміру програмних систем під управлінням платформи WordPress.

Вимоги до програми:

Вимоги функціональних характеристик.

Вимоги до складу функцій, що виконуються.

Програма має виконувати такі функції:

1) Вводити з інтерфейсу програми три метрики діаграми класів, за якими потрібно оцінити розмір ПС, створених для системи керування вмістом WordPress на мові програмування PHP в тисячах строк коду: друга , третя 1 X 2 та четверта X 3 метрики визначають відповідно загальну кількість класів, середню кількість методів на клас та глибину дерева успадкування.

2) Виконувати обчислення раннього прогнозування розміру програмних систем під управлінням платформи WordPress.

3) Виконувати обчислення довірчого інтервалу нелінійної регресії.

4) Виконувати обчислення інтервалу передбачення нелінійної регресії.

Вимоги до організації вхідних та вихідних даних

Вхідні дані:

- три метрики діаграми класів, за якими потрібно для раннього прогнозування розміру програмних систем під управлінням платформи WordPress: загальна кількість класів, кількість методів на клас та глибина дерева успадкування.

Вихідні дані:

- значення границь довірчого інтервалу нелінійної регресії та границь інтервалу передбачення нелінійної регресії раннього прогнозування розміру програмних систем під управлінням платформи WordPress.

Вимоги до надійності не висуваються.

Вимоги до умов експлуатації не висуваються.

Вимоги до складу й параметрів технічних засобів.

Програма має експлуатуватись на обладнанні з параметрами не нижчими за наступні:

64-бітний процесор на архітектурі x86 або ARM, оперативна пам’ять 2 Гб, простір на жорсткому диску 800 Мб.

Вимоги до інформаційної та програмної сумісності.

Для запуску та роботи з програмним забезпеченням необхідна операційна система Windows 7/8/10/11.

Вимоги до маркування та пакування.

Визначаються вимогами до пакування CD дисків.

Вимоги до транспортування та зберігання.

Програма може зберігатись та транспортуватися на CD дисках.

Вимоги до програмної документації.

Програмна документація має включати: технічне завдання на розробку програми, текст програми, опис програми, інструкцію користувача, програму та методику випробувань.

Техніко-економічні показники не розраховуються**.**

Стадії та етапи розробки.

Стадії та етапи розробки програми представлені у таблиці А.1.

Таблиця А.1- Стадії та етапи розробки.

Стадія розробки	Етапи розробки	Дата початку	Дата завершення
Технічне завдання	Розробка та затвердження технічного завдання	11.09.2025	14.09.2025
Ескізний проєкт	Розробка ескізного проєкту	15.09.2025	23.09.2025
	Затвердження ескізного проєкту	24.09.2025	25.09.2025
Технічний проєкт	Розробка технічного проєкту	26.09.2025	07.10.2025
	Затвердження технічного проєкту	08.10.2025	11.10.2025
Робочий проєкт	Розробка програмної документації	12.10.2025	22.10.2025
	Випробування програми	23.10.2025	17.11.2025

Порядок контролю та приймання

Порядок контролю та приймання програми здійснюється представниками замовника у присутності представника розробника.

Для приймання програми надається програмна документація, яка була розроблена відповідно до технічного завдання. Представники замовника установлюють відповідність програми технічному завданню.

За результатами приймання програми представниками замовника складається акт приймання програми.

ДОДАТОК Б - КОД ПРОГРАМНОЇ СИСТЕМИ «INFOPLUG» РАННЬОГО ПРОГНОЗУВАННЯ РОЗМІРУ ПС ПІД УПРАВЛІННЯМ ПЛАТФОРМИ WORDPRESS

Interface.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>

<BorderPane xmlns:fx="http://javafx.com/fxml/1"
        fx:controller="com.example.kloc.Controller">
   <center>
      <HBox spacing="12" padding="12">
         <children>
            <VBox spacing="8">
               <children>
                  <TitledPane text="Input" collapsible="false">
                     <content>
                        <GridPane hgap="8" vgap="8" padding="8">
                           <columnConstraints>
                              <ColumnConstraints percentWidth="40"/>
                              <ColumnConstraints percentWidth="60"/>
                           </columnConstraints>

                           <Label text="Class count:" GridPane.rowIndex="0" GridPane.columnIndex="0"/>
                           <TextField fx:id="classField" GridPane.rowIndex="0" GridPane.columnIndex="1"/>

                           <Label text="Avg methods count:" GridPane.rowIndex="1"
GridPane.columnIndex="0"/>
                           <TextField fx:id="methodsField" GridPane.rowIndex="1" GridPane.columnIndex="1"/>

                           <Label text="Depth of inheritance tree (DIT):" GridPane.rowIndex="2"
GridPane.columnIndex="0"/>
                           <TextField fx:id="ditField" GridPane.rowIndex="2" GridPane.columnIndex="1"/>
```

```
                    <Button text="Calculate" onAction="#onCalculate" GridPane.rowIndex="3"
GridPane.columnIndex="1"/>
                </GridPane>
            </content>
        </TitledPane>
    </children>
</VBox>

<VBox spacing="8">
    <children>
        <TitledPane text="Result" collapsible="false">
            <content>
                <GridPane hgap="8" vgap="8" padding="8">
                    <columnConstraints>
                        <ColumnConstraints percentWidth="40"/>
                        <ColumnConstraints percentWidth="60"/>
                    </columnConstraints>

                    <Label text="KLOC:" GridPane.rowIndex="0" GridPane.columnIndex="0"/>
                    <TextField fx:id="klocField" editable="false" GridPane.rowIndex="0"
GridPane.columnIndex="1"/>

                    <Label text="Confidence interval:" GridPane.rowIndex="1"
GridPane.columnIndex="0"/>
                    <TextField fx:id="confIntervalField" editable="false" GridPane.rowIndex="1"
GridPane.columnIndex="1"/>

                    <Label text="Prediction interval:" GridPane.rowIndex="2" GridPane.columnIndex="0"/>
                    <TextField fx:id="predIntervalField" editable="false" GridPane.rowIndex="2"
GridPane.columnIndex="1"/>
                </GridPane>
            </content>
        </TitledPane>
    </children>
</VBox>
    </children>
</HBox>
```

```
    </center>
</BorderPane>
```

Main.java

```
package com.example.kloc;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.stage.Stage;

public class Main extends Application {
    @Override
    public void start(Stage stage) throws Exception {
        FXMLLoader loader = new FXMLLoader(getClass().getResource("/com/example/kloc/layout.fxml"));
        Scene scene = new Scene(loader.load());
        stage.setTitle("KLOC Calculator");
        stage.setScene(scene);
        stage.setWidth(650);
        stage.setHeight(300);
        stage.show();
    }

    public static void main(String[] args) {
        launch();
    }
}
```

ModelConfig.java

```
package com.example.kloc;

public class ModelConfig {

    public static final double INTERCEPT = -1.6534196588628516;

    public static final double[] COEFFS = {
            1.00985263e+00,
```

```
            1.03356171e+00,
            3.77566188e-04
    };

    public static final int N = 108;

    public static final double MSE = 0.007749313690538087;

    public static final double[][] ZX = {
            { 0.03440772, 0.04800593, -0.07265221},
            { 0.04800593, 0.35744514, 0.06761312},
            {-0.07265221, 0.06761312, 4.17659552}
    };

    public static final double[] Z_MEAN = {1.712661, 0.956846, 0.083818};
}
```

KlocCalculator.java

```
package com.example.kloc;

import org.apache.commons.math3.distribution.TDistribution;

public class KlocCalculator {

    public static class Result {
        public double value;
        public double confLower;
        public double confUpper;
        public double predLower;
        public double predUpper;
    }

    public Result calculate(double classCount, double methodsCount, double dit) {

        Result r = new Result();

        // 1) Основне рівняння
        double value = Math.pow(10.0, ModelConfig.INTERCEPT)
```

```
        * Math.pow(classCount, ModelConfig.COEFFS[0])
        * Math.pow(methodsCount, ModelConfig.COEFFS[1])
        * Math.pow(dit, ModelConfig.COEFFS[2]);

r.value = value;

// 2) Обчислення Z-векторів
double[] logs = {
        Math.log10(classCount),
        Math.log10(methodsCount),
        Math.log10((dit > 0) ? dit : 1e-12)
};

double[] zx = new double[3];
for (int i = 0; i < 3; i++) zx[i] = logs[i] - ModelConfig.Z_MEAN[i];

// ZX * zx
double[] temp = new double[3];
for (int i = 0; i < 3; i++) {
    double s = 0.0;
    for (int j = 0; j < 3; j++) {
        s += ModelConfig.ZX[i][j] * zx[j];
    }
    temp[i] = s;
}

// zx^T * ZX * zx
double se_z = 0.0;
for (int i = 0; i < 3; i++) se_z += zx[i] * temp[i];

int dfT = ModelConfig.N - 4;
TDistribution tDist = new TDistribution(dfT);
double tQuantile = tDist.inverseCumulativeProbability(1 - (1 - 0.95) / 2.0);

double pred_val = tQuantile * Math.sqrt(ModelConfig.MSE * (1.0 + 1.0 / ModelConfig.N + se_z));
double conf_val = tQuantile * Math.sqrt(ModelConfig.MSE * (1.0 / ModelConfig.N + se_z));

// інтервали
```

```
        double logValue = Math.log10(value);
        r.predLower = Math.pow(10.0, logValue - pred_val);
        r.predUpper = Math.pow(10.0, logValue + pred_val);

        r.confLower = Math.pow(10.0, logValue - conf_val);
        r.confUpper = Math.pow(10.0, logValue + conf_val);

        return r;
    }
}
```

Controller.java

```
package com.example.kloc;


import javafx.fxml.FXML;
import javafx.scene.control.TextField;
import java.text.DecimalFormat;


public class Controller {


    @FXML private TextField classField;
    @FXML private TextField methodsField;
    @FXML private TextField ditField;


    @FXML private TextField klocField;
    @FXML private TextField confIntervalField;
    @FXML private TextField predIntervalField;


    private final DecimalFormat df = new DecimalFormat("#0.0000");


    private final KlocCalculator calculator = new KlocCalculator();


    @FXML
    private void onCalculate() {
```

```
        try {
            double classCount = Double.parseDouble(classField.getText().trim().replace(",", "."));
            double methodsCount = Double.parseDouble(methodsField.getText().trim().replace(",", "."));
            double dit = Double.parseDouble(ditField.getText().trim().replace(",", "."));

            KlocCalculator.Result r = calculator.calculate(classCount, methodsCount, dit);

            klocField.setText(df.format(r.value));
            confIntervalField.setText(df.format(r.confLower) + " - " + df.format(r.confUpper));
            predIntervalField.setText(df.format(r.predLower) + " - " + df.format(r.predUpper));

        } catch (Exception ex) {
            klocField.setText("Invalid input");
            confIntervalField.clear();
            predIntervalField.clear();
        }
    }
}
```

ДОДАТОК В - ОПИС ПРОГРАМНОЇ СИСТЕМИ «INFOPLUG» РАННЬОГО ПРОГНОЗУВАННЯ РОЗМІРУ ПС ПІД УПРАВЛІННЯМ ПЛАТФОРМИ WORDPRESS

Загальні відомості

Програма "Infoplug" для раннього прогнозування розміру програмних систем під управлінням платформи WordPress.

Функціональне призначення

Програма "Infoplug" призначена для раннього прогнозування розміру програмних систем під управлінням платформи WordPress.

Технічні засоби, що використовуються

Програма має експлуатуватись на обладнанні з параметрами не нижчими за наступні: 64-бітний процесор на архітектурі x86 або ARM, оперативна пам’ять 2 Гб, простір на жорсткому диску 800 Мб.

Виклик та завантаження

Для запуску та роботи з програмним забезпеченням необхідна операційна система Windows 7/8/10/11.

Завантаження програми здійснюється шляхом відкриття файлу Infoplug.exe.

ДОДАТОК Г - ІНСТРУКЦІЯ КОРИСТУВАЧА ПРОГРАМНОЇ СИСТЕМИ «INFOPLUG» РАННЬОГО ПРОГНОЗУВАННЯ РОЗМІРУ ПС ПІД УПРАВЛІННЯМ ПЛАТФОРМИ WORDPRESS

Загальний опис роботи програмної системи

Після завантаження файлу Infoplug.exe на екрані з’явиться форма для відображення початкового стану програми для раннього прогнозування розміру програмних систем під управлінням платформи WordPress (рисунок Г.1).

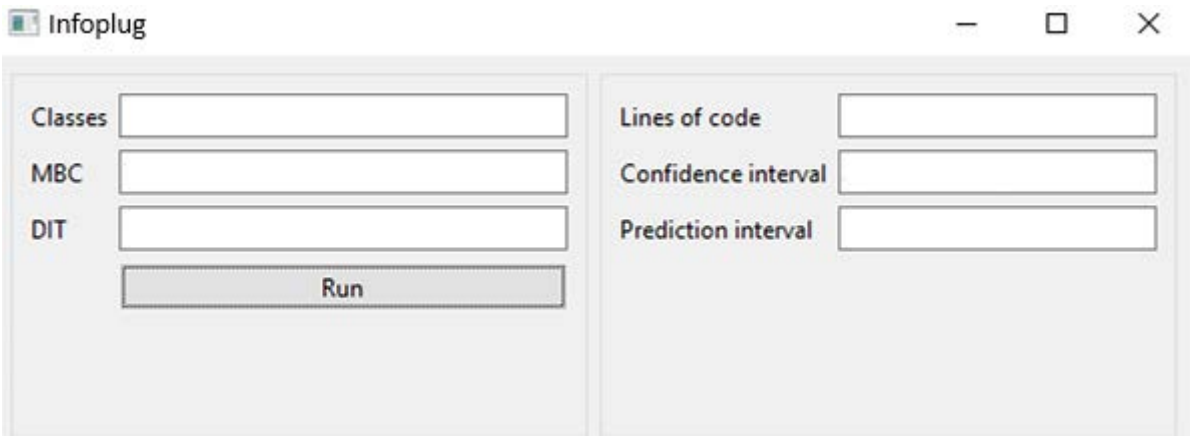

Рисунок Г.1 – Форма для відображення початкового стану програми

У графічному вікні (див. рис. Г.1) у відповідних рядках редагування потрібно ввести дані: загальну кількість класів, загальну кількість методів на кожен клас, глибину дерева успадкування та натиснути кнопку «Run», після чого з’явиться результат прогнозування розміру ПС та значення границь довірчого інтервалу та інтервалу прогнозування.

ДОДАТОК Д - ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ ПРОГРАМНОЇ СИСТЕМИ «INFOPLUG» РАННЬОГО ПРОГНОЗУВАННЯ РОЗМІРУ ПС ПІД УПРАВЛІННЯМ ПЛАТФОРМИ WORDPRESS

Об’єкт випробувань: Програма для раннього прогнозування розміру програмних систем під управлінням платформи WordPress.

Ціль випробувань: перевірка відповідності програми Infoplug до вимог технічного завдання і повної реалізації.

Склад програмної документації:

− технічне завдання;

− інструкція користувача;

− опис програми;

− текст програми;

− програма та методика випробувань

Технічні вимоги.

Програма має експлуатуватись на обладнанні з параметрами не нижчими за наступні: 64-бітний процесор на архітектурі x86 або ARM, оперативна пам’ять 2 Гб, простір на жорсткому диску 800 Мб.

Методи випробувань.

Випробування програмного забезпечення проводиться представником замовника у присутності розробника. Згідно з описом, здійснюється запуск програми "Infoplug", після чого проводиться перевірка її працездатності. Переглядаються всі екранні форми та виконуються всі дозволені операції відповідно до посібника користувача. Якщо програма коректно виконує свої функції відповідно до вимог технічного завдання та супровідної документації, розробку вважають завершеною та переходять до етапу впровадження програмного продукту в експлуатацію. У разі виявлення невідповідностей програму повертають розробнику на доопрацювання.

Для випробування застосовуються методи відповідно до концепції «чорної скриньки» (див. рис. Д.1).

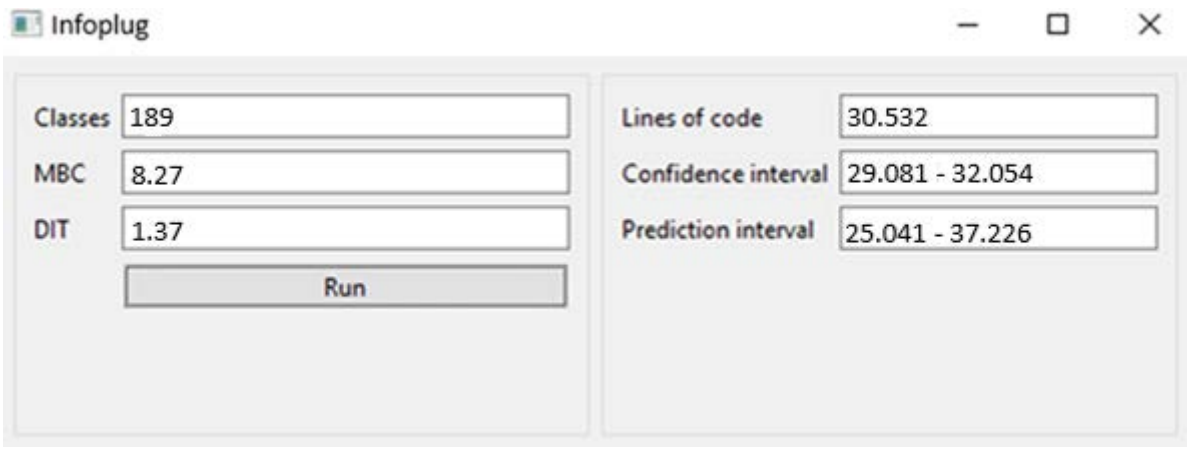

Рисунок Д.1 – Графічне вікно програми на платформі Windows 11