

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра програмного забезпечення автоматизованих систем
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма «Інженерія програмного забезпечення»

«Допущений до захисту»
Завідувач кафедри

«__» _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти «магістр»

на тему: Нелінійна регресійна модель для прогнозування розміру мікросервісних e-commerce додатків на Java та розробка програмного забезпечення для її реалізації

Виконав: студент групи 6151м

 Алієв А.Ш.
(підпис, ПІБ)

Керівник роботи:

доцент кафедри ПЗАС, к.т.н., доцент
(посада, науковий ступень вчене звання)

 Пухалевич А.В.
(підпис, ПІБ)

Миколаїв – 2024 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
 Кафедра програмного забезпечення автоматизованих систем
 Спеціальність 121 «Інженерія програмного забезпечення»
 Освітня програма «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»

Гарант освітньої програми

_____ проф. С.Б.Приходько
 (підпис)

« 14 » жовтня 2024 р.

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

Студенту Алієву Анару Шихалі Огли

(Прізвище, ім'я, по батькові)

1. Тема роботи: Нелінійна регресійна модель для прогнозування розміру мікросервісних e-commerce додатків на Java та розробка програмного забезпечення для її реалізації

Керівник роботи Пухалевич Андрій Володимирович, к.т.н., доцент

Затверджені наказом ректора № 1070-УЧ від « 11 » 10 2024 р.

2. Термін подання роботи: 09.12.2024 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; Розділ з розробки 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 1.Тема 2. Актуальність теми, 3. Мета і завдання дослідження, 4. Об'єкт і предмет дослідження, 5. Методи дослідження, 6. Наукова новизна одержаних результатів, 7. Практичне значення одержаних результатів, 8. Особистий внесок здобувача 9. Апробація результатів досліджень та публікації, 10. Аналіз існуючих підходів до прогнозування розміру програмних проектів на Java 11. Визначення доцільності проведення дослідження 12. Вихідні дані та їх передобробка 13. Нелінійна регресійна модель 14. Лінійна регресійна модель 15. Порівняльний аналіз моделей, 16. Контекстна діаграма ПЗ, 17. Декомпозиція контекстного процесу. 18. Концептуальна модель даних, 19. Модель переходів станів, 20. Графічний інтерфейс. 21.Висновки.

6. Консультанти розділів роботи

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

7. Дата видачі завдання 14.10.2024 р.

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи	Терміни виконання	Примітка
1.	Підготовка розділу ВСТУП	16.10.2024	НС*
2.	Підготовка розділу з огляду літератури та обґрунтування необхідності проведення досліджень за обраною темою	18.10.2024	НС*
3.	Підготовка розділу (ів) за результатами власних досліджень	21.10.2024	НС*
4.	Підготовка розділу з розробки програмного забезпечення	27.11.2024	
5.	Підготовка організаційно-економічного розділу	29.11.2024	
6.	Підготовка розділу з охорони праці	02.12.2024	
7.	Підготовка розділу з охорони навколишнього середовища	04.12.2024	
8.	Підготовка розділу ВИСНОВКИ	05.12.2024	
9.	Оформлення списку використаних джерел та додатків	06.12.2024	
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.)	09.12.2024	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку)
11.	Підготовка презентації та доповіді	13.12.2024	
12.	Попередній захист роботи на засіданні кафедри ПЗАС	16.12.2024	
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	19.12.2024	

* - за результатами наукового стажування (НС), яке було з 02.09.2024 по 13.10.2024 р.

Студент


(підпис)

А.Ш. Алієв

(ПІБ)

Керівник роботи


(підпис)

А.В. Пухалевич

(ПІБ)

РЕФЕРАТ

Алієв Анар Шихалі огли

Нелінійна регресійна модель для прогнозування розміру мікросервісних e-commerce додатків на Java та розробка програмного забезпечення для її реалізації

Кваліфікаційна робота на здобуття освітнього рівня магістра зі спеціальності 121 – «Інженерія програмного забезпечення». Національний університет кораблебудування імені адмірала Макарова. Миколаїв, 2024 р.

Обсяг роботи: 100 стор., 13 табл., 12 рис., 21 використане джерело, 5 додатків.

Актуальність теми. Прогнозування розміру програмних проєктів на ранніх стадіях розробки допомагає в управлінні ресурсами, плануванні часу, бюджету та в зменшенні ризиків. Кращу якість мають моделі, що побудовані під певні архітектуру, мову програмування, фреймворк, тип програмного забезпечення.

Для e-commerce додатків на Java, що розробляються на базі мікросервісів, моделі не знайдено. Проте у сучасному світі електронна комерція швидко розвивається, що потребує нових архітектурних рішень. Одним із таких рішень є використання мікросервісів, тому актуальною є задача побудови двохфакторної нелінійної регресійної моделі для прогнозування розміру мікросервісних e-commerce додатків на Java.

Мета дослідження. Мета дослідження полягає в підвищенні достовірності прогнозування розміру e-commerce додатків на Java, що розробляються на базі мікросервісів, шляхом побудови двохфакторної нелінійної регресійної моделі.

Об'єкт дослідження. В якості об'єкту дослідження виступає процес прогнозування розміру мікросервісних e-commerce додатків на Java.

Предмет дослідження. Предметом дослідження є нелінійні регресійні моделі прогнозування розміру e-commerce додатків на Java, що розробляються на базі мікросервісів.

Методи дослідження. При дослідженні використовувалися методи теорії ймовірностей, математичної статистики, лінійного та нелінійного регресійного аналізу.

Наукова новизна одержаних результатів. Виконано удосконалення нелінійної двохфакторної регресійної моделі прогнозування розміру e-commerce додатків на Java, що розробляються на базі мікросервісів, за рахунок використання нормалізуючого перетворення десятковий логарифм та вилучення багатовимірних

викидів. Це дало змогу підвищити достовірність даного прогнозування в порівнянні з побудованою двохфакторною лінійною регресійною моделлю.

Практичне значення отриманих результатів В кваліфікаційній роботі програмно реалізована модель для прогнозування розміру e-commerce додатків на Java, що розробляються на базі мікросервісів. Це дає змогу автоматизувати, скоротити час та підвищити достовірність прогнозування на ранніх етапах роботи над проектом.

Апробація результатів досліджень. Результати досліджень, що викладені в роботі, були представлені на VII Всеукраїнській науково-практичній інтернет конференції студентів, аспірантів та молодих вчених «Сучасні інформаційні системи і технології» (29 листопада 2024 р., м. Херсон, м. Хмельницький).

Публікації. Результати досліджень викладені в 1 роботі.

Ключові слова: мікросервіси, e-commerce додатки, Java, прогнозування розміру, нормалізуюче перетворення, нелінійна регресія.

ABSTRACT

Anar Aliiev

A nonlinear regression model for predicting the size of microservice e-commerce applications in Java and software developing for its implementation

Qualification work for obtaining a master's degree in Specialty 121 – Software Engineering. Admiral Makarov National University of Shipbuilding. Mykolaiv, 2024.

The work contains: 100 pages, 13 tables, 12 figures, 21 references, 5 appendices.

Relevance of the theme. Forecasting the size of software projects in the early stages of development helps in resource management, time planning, budgeting and risk reduction. Models that are built for a specific architecture, programming language, framework, software type have the best quality.

No model has been found for e-commerce applications in Java, which are developed based on microservices. However, in the modern world, e-commerce is developing rapidly, which requires new architectural solutions. One of such solutions is the use of microservices. Therefore, the task of building a two-factor nonlinear regression model for predicting the size of microservice e-commerce applications in Java is relevant.

Purpose of the research. The purpose of the study is to increase the reliability of predicting the size of e-commerce applications in Java developed on the basis of microservices by building a two-factor nonlinear regression model.

The object of the study. The object of the study is the process of predicting the size of microservice e-commerce applications in Java.

The subject of the study. The subject of the study is nonlinear regression models for predicting the size of e-commerce applications in Java, developed based on microservices.

Research methods. The research used methods of probability theory, mathematical statistics, linear and nonlinear regression analysis.

Scientific novelty of the obtained results. The nonlinear two-factor regression model for predicting the size of e-commerce applications in Java developed on the basis of microservices was improved by using the normalizing transformation of the decimal logarithm and removing multivariate outliers. This made it possible to increase the reliability of this prediction in comparison with the constructed two-factor linear regression model.

The practical significance of the obtained results. The qualification work software implements a model for predicting the size of e-commerce applications in Java, developed on the basis of microservices. This allows you to automate, reduce time and increase the reliability of forecasting in the early stages of work on the project.

Approbation of research results. The results of the research presented in the work were presented at the VII All-Ukrainian Scientific and Practical Internet Conference of Students, Postgraduate Students and Young Scientists "Modern Information Systems and Technologies" (November 29, 2024, Kherson, Khmelnytskyi).

Publications. The research results are presented in 1 paper.

Keywords: мікросервіси, e-commerce додатки, Java, прогнозування розміру, нормалізуюче перетворення, нелінійна регресія.

ЗМІСТ

ВСТУП.....	10
1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ТА МАТЕМАТИЧНИХ МОДЕЛЕЙ ДЛЯ РАНЬОГО ПРОГНОЗУВАННЯ РОЗМІРУ ПРОГРАМНИХ ПРОЕКТІВ НА JAVA ТА ОБГРУНТУВАННЯ НЕОБХІДНОСТІ ДОСЛІДНИЦЬКОЇ РОБОТИ ЗА ОБРАНОЮ ТЕМОЮ	13
1.1 Аналіз існуючих методів та математичних моделей для прогнозування розміру програмних проєктів на Java	13
1.2 Оцінювання якості регресійної моделі для прогнозування розміру програмних проєктів	18
1.3 Обґрунтування необхідності дослідницької роботи за обраною темою	19
1.4 Висновки до розділу 1	20
2 НЕЛІНІЙНА РЕГРЕСІЙНА МОДЕЛЬ ДЛЯ ПРОГНОЗУВАННЯ РОЗМІРУ МІКРОСЕРВІСНИХ E-COMMERCE ДОДАТКІВ НА JAVA	21
2.1 Аналіз вихідних даних на мультиколінеарність та викиди	21
2.2 Розробка двохфакторної нелінійної регресійної моделі для прогнозування розміру e-commerce додатків на Java	26
2.3 Висновки до розділу 2	32
3 ПРОЕКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПРОГНОЗУВАННЯ РОЗМІРУ МІКРОСЕРВІСНИХ E-COMMERCE ДОДАТКІВ НА JAVA	34
3.1 Постановка задачі на розробку ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java	34
3.2 Ескізний проєкт програмного забезпечення для раннього оцінювання розміру мікросервісів на Java, що створюються з використанням фреймворку Spring	35
3.2.1 Контекстна діаграма ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java	35
3.2.2 Концептуальна модель даних ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java	38
3.2.3 Модель переходів станів ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java	39

3.2.4 Проект графічного інтерфейсу ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java	40
3.3 Технічний проект ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java	42
3.3.1 Декомпозиція контекстного процесу ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java до рівня специфікацій програмних модулів	43
3.3.2 Логічна модель даних ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java	48
3.4 Робочий проект ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java	49
3.4.1 Вибір мови програмування та засобів розробки ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java	49
3.4.2 Вибір засобів розробки ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java	50
3.4.3 Випробування ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java	53
3.5 Висновки до розділу 3	54
4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ РОЗРОБКИ ТА ВПРОВАДЖЕННЯ ПЗ ДЛЯ ПРОГНОЗУВАННЯ РОЗМІРУ МІКРОСЕРВІСНИХ E-COMMERCE ДОДАТКІВ НА JAVA	55
4.1 Вступ	55
4.2 Розрахунок витрат на створення й експлуатацію програми для прогнозування розміру мікросервісних e-commerce додатків на Java	55
4.3 Економічна ефективність розробки і впровадження ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java	58
4.4 Висновки до розділу 4	59
5 ОХОРОНА ПРАЦІ	60
5.1 Вступ	60
5.2 Аналіз шкідливих та небезпечних факторів на робочому місці програміста	60
5.3 Техніка безпеки при роботі на комп'ютері	66
6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА	71

6.1 Аналіз існуючої проблеми необхідності охорони навколишнього середовища...	71
6.2 Забруднення навколишнього середовища офісом комп'ютерної фірми	73
6.3 Розробка заходів щодо зменшення забруднення	75
ВИСНОВКИ	78
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	79
ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ ПЗ ДЛЯ ПРОГНОЗУВАННЯ РОЗМІРУ МІКРОСЕРВІСНИХ Е-COMMERCE ДОДАТКІВ НА JAVA	82
ДОДАТОК Б – ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ ПЗ ДЛЯ ПРОГНОЗУВАННЯ РОЗМІРУ МІКРОСЕРВІСНИХ Е-COMMERCE ДОДАТКІВ НА JAVA	86
ДОДАТОК В – ОПИС ПЗ ДЛЯ ПРОГНОЗУВАННЯ РОЗМІРУ МІКРОСЕРВІСНИХ Е-COMMERCE ДОДАТКІВ НА JAVA	89
ДОДАТОК Г – ІНСТРУКЦІЯ КОРИСТУВАЧА ПЗ ДЛЯ ПРОГНОЗУВАННЯ РОЗМІРУ МІКРОСЕРВІСНИХ Е-COMMERCE ДОДАТКІВ НА JAVA	91
ДОДАТОК Д – ТЕКСТ ПЗ ДЛЯ ПРОГНОЗУВАННЯ РОЗМІРУ МІКРОСЕРВІСНИХ Е-COMMERCE ДОДАТКІВ НА JAVA	94

ВСТУП

Актуальність теми. Прогнозування розміру програмних проектів на ранніх стадіях розробки допомагає в управлінні ресурсами, плануванні часу, бюджету та в зменшенні ризиків. Виходячи з цього в наш час велика увага приділяється підвищенню достовірності такого прогнозування, що призводить до розробки нових та вдосконалення існуючих методів і моделей.

В ході аналізу публікацій за темою кваліфікаційної роботи виявлено, що для проектів на Java для оцінювання розміру за об'єктно-орієнтованими метриками побудовано як лінійні, так і нелінійні регресійні моделі, які в свою чергу є однофакторними або багатофакторними [1-7]. Використання лінійної регресійної теорії справедливо лише в окремих випадках, тому, в основному, будувалися нелінійні регресійні моделі. В переважній більшості робіт розроблено моделі множинної регресії з різними наборами незалежних факторів, так як відомо, що багатофакторні моделі зазвичай мають кращу якість. Також із аналізу публікацій випливає, що кращу якість мають моделі, що побудовані під певні архітектуру, мову програмування, фреймворк, тип програмного забезпечення [8, 9].

Для e-commerce додатків на Java, розроблених на базі мікросервісів, моделі не знайдено. Проте у сучасному світі електронна комерція швидко розвивається, збільшується обсяг даних, ускладнюються системи, що потребує нових архітектурних рішень. Одним із таких рішень є використання мікросервісів, що дає можливість підвищити надійність, швидкість розробки, швидкість та ціну розгортання, зменшити вартість підтримки. Тому актуальною є задача побудови двохфакторної нелінійної регресійної моделі для прогнозування розміру мікросервісних e-commerce додатків на Java.

Мета і завдання дослідження. Мета дослідження полягає в підвищенні достовірності прогнозування розміру e-commerce додатків на Java, що розробляються на базі мікросервісів, шляхом побудови двохфакторної

нелінійної регресійної моделі. Для досягнення мети дослідження необхідно вирішити наступні завдання:

- Провести аналіз наявних методів і моделей для прогнозування розміру e-commerce додатків на Java, що розробляються на базі мікросервісів, на ранніх етапах роботи над проектом.
- Виконати збір емпіричних даних для побудови двохфакторної нелінійної регресійної моделі прогнозування розміру e-commerce додатків на Java, що розроблені на базі мікросервісів.
- Перевірити потенційні фактори на мультиколінеарність.
- Перевірити емпіричні дані на відповідність гаусівському розподілу.
- Виконати нормалізацію вихідних даних за допомогою десятичного логарифму.
- Вилучити викиди в даних, використовуючи квадрат відстані Махаланобіса.
- Побудувати двохфакторну лінійну регресійну модель з обґрунтуванням правомірності її побудови.
- Побудувати довірчі інтервали та інтервали прогнозування для лінійної регресійної моделі.
- Побудувати, з використанням зворотнього перетворення, двохфакторну нелінійну регресійну модель прогнозування розміру e-commerce додатків на Java, що розробляються на базі мікросервісів, її довірчі інтервали та інтервали прогнозування, забезпечивши при цьому належність розміру з емпіричних даних інтервалам прогнозування.
- Оцінити якість отриманої двохфакторної нелінійної регресійної моделі.
- Виконати програмну реалізацію двохфакторної нелінійної регресійної моделі прогнозування розміру e-commerce додатків на Java, що розробляються на базі мікросервісів.

Об’єкт дослідження. В якості об’єкту дослідження виступає процес прогнозування розміру мікросервісних e-commerce додатків на Java.

Предмет дослідження. Предметом дослідження є нелінійні регресійні моделі прогнозування розміру e-commerce додатків на Java, що розробляються на базі мікросервісів.

Методи дослідження. При дослідженні використовувалися методи теорії ймовірностей, математичної статистики, лінійного та нелінійного регресійного аналізу.

Наукова новизна одержаних результатів. Виконано удосконалення нелінійної двохфакторної регресійної моделі прогнозування розміру e-commerce додатків на Java, що розробляються на базі мікросервісів, за рахунок використання нормалізуючого перетворення десятковий логарифм та вилучення багатовимірних викидів. Це дало змогу підвищити достовірність даного прогнозування в порівнянні з розробленою лінійною двохфакторною регресійною моделлю прогнозування розміру e-commerce додатків на Java, що розробляються на базі мікросервісів.

Практичне значення отриманих результатів. В кваліфікаційній роботі програмно реалізована модель для прогнозування розміру e-commerce додатків на Java, що розробляються на базі мікросервісів. Це дає змогу автоматизувати, скоротити час та підвищити достовірність прогнозування на ранніх етапах роботи над проектом.

Особистий внесок здобувача. Основні результати кваліфікаційної роботи отримані здобувачем самостійно. У тезах конференції [10], що опубліковані у співавторстві, здобувачу належить розробка нелінійної двохфакторної регресійної моделі прогнозування розміру e-commerce додатків на Java, що розробляються на базі мікросервісів.

Апробація результатів роботи. Результати досліджень, що викладені в роботі, були представлені на VII Всеукраїнській науково-практичній інтернет конференції студентів, аспірантів та молодих вчених «СУЧАСНІ ІНФОРМАЦІЙНІ СИСТЕМИ І ТЕХНОЛОГІЇ» (29 листопада 2024 р., м. Херсон, м. Хмельницький).

Публікації. Результати досліджень опубліковані 1 роботі.

1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ТА МАТЕМАТИЧНИХ МОДЕЛЕЙ ДЛЯ РАНЬОГО ПРОГНОЗУВАННЯ РОЗМІРУ ПРОГРАМНИХ ПРОЕКТІВ НА JAVA ТА ОБГРУНТУВАННЯ НЕОБХІДНОСТІ ДОСЛІДНИЦЬКОЇ РОБОТИ ЗА ОБРАНОЮ ТЕМОЮ

1.1 Аналіз існуючих методів та математичних моделей для прогнозування розміру програмних проектів на Java

Прогнозування розміру програмних проектів на ранніх стадіях розробки допомагає в управлінні ресурсами, плануванні часу, бюджету та в зменшенні ризиків. Виходячи з цього в наш час велика увага приділяється підвищенню достовірності такого прогнозування, що призводить до розробки нових та вдосконалення існуючих методів і моделей.

В результаті аналізу публікацій за темою дослідження були виявлені різні групи метрик розміру програмних застосунків, основними з яких є: кількість рядків коду, кількість функціональних точок, метрики Холстеда та інші [11, 12]. Також існують різноманітні групи методів та математичних моделей для раннього прогнозування розміру програмних додатків [13, 14], серед яких експертне оцінювання, параметричне оцінювання, регресійний аналіз, нейронні мережі, комбіновані методи та моделі. У всіх них є певні переваги, недоліки та сфери застосування.

Для подальшого дослідження обрано регресійний аналіз. Виявлено, що для проектів на Java для оцінювання розміру за об'єктно-орієнтованими метриками побудовано як лінійні, так і нелінійні регресійні моделі, які в свою чергу є однофакторними або багатфакторними [1-7]. Використання лінійної регресійної теорії справедливо лише в окремих випадках, тому, зазвичай, будувалися нелінійні регресійні моделі. В переважній більшості робіт розроблено моделі множинної регресії з різними наборами незалежних

факторів, так як відомо, що багатфакторні моделі в більшості випадків мають кращу якість.

В [2, 3] представлено лінійні регресійні рівняння для оцінювання розміру інформаційних систем на Java та PHP. У якості факторів та відгуку використовувалися наступні метрики концептуальної моделі даних у вигляді діаграми класів: загальна кількість сутностей, загальна кількість відносин, загальна кількість атрибутів, число рядків коду (KLOC). В [5] розроблена однофакторна нелінійна регресійна модель для оцінювання розміру веб-додатків на Java з використанням одновимірного нормалізуючого перетворення Джонсона. В якості залежної змінної використовується число рядків коду (KLOC), незалежної – кількість класів. В [4] розроблена нелінійна регресійна модель для оцінювання розміру програмного забезпечення систем з відкритим кодом на Java, яка містить 3 фактори. При побудові моделі використовувалося багатовимірне перетворення Джонсона для сімейства SB та наступні метрики: кількість рядків коду (KLOC), загальна кількість класів, середня кількість атрибутів на клас та загальна кількість зв'язків. В [6], з використанням п'ятивимірного перетворення Джонсона для сімейства SB, побудовано чотирьохфакторну нелінійну регресійну модель для оцінювання розміру Java-застосунків з відкритим кодом. Для побудови моделі використовувалися метрики, що визначаються за діаграмою класів: кількість строк коду, кількість класів, метрика, що характеризує відсутність згуртованості методів, кількість викликів унікального методу в класі і кількість статичних методів.

Із аналізу публікацій випливає, що кращу якість мають моделі, що побудовані під певні архітектуру, мову програмування, фреймворк, тип програмного забезпечення [8, 9]. Для e-commerce додатків на Java, розроблених на базі мікросервісів, не знайдено моделі для прогнозування розміру проектів на ранніх стадіях розробки. Тому в кваліфікаційній роботі буде створена двохфакторна нелінійна регресійна модель для прогнозування

розміру мікросервісних e-commerce додатків на Java з використанням нормалізуючого перетворення у вигляді десяткового логарифму.

Якщо емпіричні дані $Y, X_1, X_2, \dots, X_k$ нормально розподілені і відсутня лінійна залежність між пояснювальними змінними $X_1, X_2, \dots, X_k$, то проблем з побудовою регресійної моделі немає. Для емпіричних даних необхідно будувати лінійну регресійну модель у вигляді [15]:

$$Y = \hat{Y} + \varepsilon = b_0 + b_1X_1 + b_2X_2 + \dots + b_kX_k + \varepsilon, \quad (1.1)$$

де $\hat{Y}$ – залежна величина,

b_i ($i = \overline{0, k}$) – параметри лінійної регресії,

X_i ($i = \overline{1, k}$) – фактори,

ε – випадкова величина з гаусівським розподілом,

k – кількість факторів.

В цьому випадку співвідношення для визначення довірчого інтервалу лінійної регресії має вигляд [15]:

$$\hat{Y} \pm t_{\alpha/2, v} S_Y \left\{ \frac{1}{N} + (\mathbf{x}^+)^T [(\mathbf{X}^+)^T \mathbf{X}^+]^{-1} (\mathbf{x}^+) \right\}^{1/2}, \quad (1.2)$$

а інтервал передбачення розраховується за формулою [4, 11]:

$$\hat{Y} \pm t_{\alpha/2, v} S_Y \left\{ 1 + \frac{1}{N} + (\mathbf{x}^+)^T [(\mathbf{X}^+)^T \mathbf{X}^+]^{-1} (\mathbf{x}^+) \right\}^{1/2}, \quad (1.3)$$

де $S_Y^2 = \frac{1}{v} \sum_{i=1}^N (Y_i - \hat{Y}_i)^2$,

$v = N - k - 1$ – кількість ступенів вільності,

N – кількість елементів в вибірці,

$t_{\alpha/2, v}$ – квантіль t -розподілу Стюдента з $\alpha/2$ рівнем значимості та v степенями вільності,

$(X^+)^T X^+$ – матриця розміром $k \times k$

$$(X^+)^T X^+ = \begin{pmatrix} S_{X_1 X_1} & S_{X_1 X_2} & \dots & S_{X_1 X_k} \\ S_{X_1 X_2} & S_{X_2 X_2} & \dots & S_{X_2 X_k} \\ \dots & \dots & \dots & \dots \\ S_{X_1 X_k} & S_{X_2 X_k} & \dots & S_{X_k X_k} \end{pmatrix}, \quad (1.4)$$

де елементи $S_{X_q X_r} = \sum_{i=1}^N [X_{qi} - \bar{X}_q][X_{ri} - \bar{X}_r]$, $q, r = \overline{1, k}$.

Якщо ж емпіричні дані $Y, X_1, X_2, \dots, X_k$ не мають нормального розподілу, то будують нелінійну регресійну модель. Одним із методів побудови нелінійної регресії є метод нормалізуючих перетворень, згідно якого необхідно [4, 6]:

- виконати нормалізацію вихідних даних, обравши нормалізуюче перетворення;
- вилучити викиди;
- побудувати лінійну регресійну модель для нормалізованих даних без викидів;
- побудувати довірчий інтервал та інтервал передбачення лінійної регресії;
- здійснити перехід до нелінійної регресійної моделі через зворотнє перетворення;
- визначити довірчий інтервал та інтервал передбачення для нелінійної моделі через зворотнє перетворення.

Позначимо нормалізуюче перетворення емпіричних даних $Y, X_1, X_2, \dots, X_k$, що не підпорядковуються гаусівському розподілу, у нормалізовані дані $Z_Y, Z_1, Z_2, \dots, Z_k$ через ψ . Тоді [15]:

$$T = \psi(P), \quad (1.5)$$

де $P = \{Y, X_1, X_2, \dots, X_k\}^T$ – емпіричні дані,

$T = \{Z_Y, Z_1, Z_2, \dots, Z_k\}^T$ – нормалізовані дані,

$\psi = \{\psi_Y, \psi_1, \psi_2, \dots, \psi_k\}^T$ – нормалізуюче перетворення.

P визначається з (1.5) через зворотнє перетворення ψ^{-1} згідно [15]:

$$P = \psi^{-1}(T). \quad (1.6)$$

Лінійна регресія для нормалізованих даних $Z_Y, Z_1, Z_2, \dots, Z_k$ має вигляд [15]:

$$Z_Y = \hat{Z}_Y + \varepsilon = b_0 + b_1 Z_1 + b_2 Z_2 + \dots + b_k Z_k + \varepsilon, \quad (1.7)$$

співвідношення для визначення довірчого інтервалу має вигляд [15]:

$$\hat{Z}_Y \pm t_{\frac{\alpha}{2}, v} S_{Z_Y} \left\{ \frac{1}{N} + (Z_X^+)^T [(Z_X^+)^T Z_X^+]^{-1} (Z_X^+) \right\}^{1/2}, \quad (1.8)$$

а співвідношення для визначення інтервалу передбачення наступне [15]:

$$\hat{Z}_Y \pm t_{\frac{\alpha}{2}, v} S_{Z_Y} \left\{ 1 + \frac{1}{N} + (Z_X^+)^T [(Z_X^+)^T Z_X^+]^{-1} (Z_X^+) \right\}^{1/2}, \quad (1.9)$$

де $S_{Z_Y}^2 = \frac{1}{v} \sum_{i=1}^N (Z_{Y_i} - \hat{Z}_{Y_i})^2$.

$v = N - k - 1$,

$(Z_X^+)^T Z_X^+$ – матриця розмірності $k \times k$ [15]:

$$(Z_X^+)^T Z_X^+ = \begin{pmatrix} S_{Z_1 Z_1} & S_{Z_1 Z_2} & \dots & S_{Z_1 Z_k} \\ S_{Z_1 Z_2} & S_{Z_2 Z_2} & \dots & S_{Z_2 Z_k} \\ \dots & \dots & \dots & \dots \\ S_{Z_1 Z_k} & S_{Z_2 Z_k} & \dots & S_{Z_k Z_k} \end{pmatrix}, \quad (1.10)$$

де елементи $S_{Z_q Z_r} = \sum_{i=1}^N [Z_{q_i} - \bar{Z}_q][Z_{r_i} - \bar{Z}_r]$, $q, r = \overline{1, k}$.

Застосувавши зворотнє перетворення (1.6) до лінійної регресійної моделі для нормалізованих даних (1.7), її довірчого інтервалу (1.8) та інтервалу

прогнозування (1.9), переходимо до нелінійної регресійної моделі для емпіричних даних, її довірчого інтервалу та інтервалу прогнозування.

За нормалізуюче перетворення обрано перетворення у вигляді десятичного логарифму.

1.2 Оцінювання якості регресійної моделі для прогнозування розміру програмних проектів

Для оцінювання якості регресійної моделі використаємо коефіцієнт детермінації R^2 , суму квадратів відхилень SS_E , середню величину відносної похибки $MMRE$ та рівень прогнозування $PRED(l)$ [15,16].

Коефіцієнт детермінації R^2 – це статистична міра того, наскільки добре модель описує виміряні дані. Значення R^2 , які ближчі до одиниці, вказують на краще підігнану модель. R^2 обчислюється по формулі [15,16]:

$$R^2 = 1 - \frac{SS_E}{SS_T}, \quad (1.11)$$

де SS_E – сума квадратів відхилень;

SS_T – загальна сума квадратів.

Сума квадратів відхилень емпіричних значень від розрахункових SS_E показує розсіювання Y по відношенню до лінії регресії і знаходиться із співвідношення [15]:

$$SS_E = \sum_{i=1}^N [y_i - \hat{y}_i]^2. \quad (1.12)$$

Для обчислення середньої величини відносної похибки $MMRE$ та рівня прогнозування $PRED(l)$ використовується величина відносної похибки MRE_i [15]:

$$MRE_i = \left| \frac{y_i - \hat{y}_i}{y_i} \right|. \quad (1.13)$$

Тоді $MMRE$ дорівнює [15]:

$$MMRE = \frac{1}{N} \sum_{i=1}^N MRE_i, \quad (1.14)$$

а $PRED(0,25)$ визначається за формулою [15]:

$$PRED(0,25) = \frac{1}{N} \sum_{i=1}^N \begin{cases} 1 & \text{if } MRE_i \leq 0.25 \\ 0 & \text{otherwise} \end{cases}. \quad (1.15)$$

Прийнятне значенням $MMRE \leq 0,25$, а $PRED(0,25) - \geq 0,75$.

1.3 Обґрунтування необхідності дослідницької роботи за обраною темою

Виходячи з вищевикладеного, можна зробити висновок про те, що регресійні моделі широко використовуються в задачах прогнозування розміру програмних проектів на ранніх етапах їх розробки.

В результаті аналізу публікацій за темою дослідження для проектів на Java для прогнозування розміру за об'єктно-орієнтованими метриками виявлено як лінійні, так і нелінійні регресійні моделі, які в свою чергу є однофакторними або багатфакторними. Використання лінійної регресійної теорії справедливо лише в окремих випадках, тому, зазвичай, будувалися нелінійні регресійні моделі. В переважній кількості робіт розроблено моделі множинної регресії з різними наборами незалежних факторів, так як відомо, що багатфакторні моделі в більшості випадків мають кращу якість.

Із аналізу публікацій випливає, що кращу якість мають моделі, що побудовані під певні архітектуру, мову програмування, фреймворк, тип програмного забезпечення.

Для e-commerce додатків на Java, розроблених на базі мікросервісів, не знайдено моделі для прогнозування розміру проєктів на ранніх стадіях розробки. Проте у сучасному світі електронна комерція швидко розвивається, збільшується обсяг даних, ускладнюються системи, що потребує нових архітектурних рішень. Одним із таких рішень є використання мікросервісів, що дає можливість підвищити надійність, швидкість розробки, швидкість та ціну розгортання, зменшити вартість підтримки. Тому затребуваною є побудова двохфакторної нелінійної регресійної моделі для прогнозування розміру мікросервісних e-commerce додатків на Java.

Все вище перераховане обґрунтовує необхідність дослідницької роботи за обраною темою. В кваліфікаційній роботі буде створена двохфакторна нелінійна регресійна модель для прогнозування розміру e-commerce додатків на Java, розроблених на базі мікросервісів, з використанням нормалізуючого перетворення у вигляді десяткового логарифму. В якості факторів обрано кількість класів та середню кількість методів, в якості залежної змінної – розмір додатку в тисячах строк коду.

1.4 Висновки до розділу 1

В даному розділі виконано аналіз існуючих методів та математичних моделей для раннього оцінювання розміру програмного забезпечення, в тому числі і на Java, та обґрунтовано необхідність проведення досліджень.

Зокрема в розділі:

- виявлено, що в даний час регресія широко використовується в задачах прогнозування;
- виявлено, що не існує регресійних моделей для прогнозування розміру e-commerce додатків на Java, розроблених на базі мікросервісів;
- виявлено, що кращу якість мають моделі, побудовані під певні архітектуру, мову програмування, фреймворк, тип програмного забезпечення.

2 НЕЛІНІЙНА РЕГРЕСІЙНА МОДЕЛЬ ДЛЯ ПРОГНОЗУВАННЯ РОЗМІРУ МІКРОСЕРВІСНИХ E-COMMERCE ДОДАТКІВ НА JAVA

2.1 Аналіз вихідних даних на мультиколінеарність та викиди

Для побудови множинної нелінійної регресійної моделі прогнозування розміру e-commerce додатків на Java використано 45 проектів вказаного типу.

Для цих проектів знято дані з наступних метрик:

- Classes (кількість класів проекту);
- MBC (середня кількість методів у класах проекту);
- KLOC (кількість строк коду проекту в тисячах).

Отримані дані наведено в таблиці 2.1.

Таблиця 2.1– Вихідні дані, що використовуються для специфікації нелінійної регресійної моделі прогнозування розміру e-commerce додатків на Java та SMD (квадрат відстані Махаланобіса)

№	Classes (X1)	MBC (X2)	KLOC (Y)	SMD
1	2	3	4	5
1	935	16,0824	76,554	2,8270
2	749	8,4032	57,446	0,7310
3	799	11,7985	51,389	2,0275
4	808	9,3218	54,876	1,3885
5	688	10,4622	47,898	0,6313
6	643	29,4774	50,642	6,9146
7	584	8,4640	38,519	1,4085
8	606	10,1568	42,641	0,6255
9	591	10,9763	45,943	0,1992
10	487	29,1951	35,865	7,2967
11	579	28,6943	38,359	8,3115
12	367	7,8965	22,707	3,5307
13	511	16,3307	34,812	1,4247
14	512	17,7031	45,075	0,6238
15	796	19,5955	71,822	3,0258

Продовження табл. 2.1

1	2	3	4	5
16	359	23,8914	39,931	3,6865
17	614	16,9463	60,428	1,9820
18	397	9,7985	24,015	2,9668
19	487	9,5708	34,857	1,2109
20	587	25,5060	62,076	5,3832
21	812	2,0530	77,94	9,0354
22	805	1,8609	61,374	3,4009
23	872	9,0115	65,673	1,3833
24	781	9,8387	52,17	1,3745
25	612	10,5915	59,389	2,2897
26	318	16,3868	26,671	2,0947
27	996	14,3725	76,061	3,0100
28	486	15,1132	42,802	0,4776
29	232	24,5129	26,862	4,9134
30	365	9,9973	39,604	3,1803
31	846	13,4728	79,515	4,4748
32	864	9,7280	66,528	1,2754
33	602	10,7757	54,096	0,9316
34	476	21,4370	60,985	7,2222
35	586	10,2679	44,939	0,3264
36	1199	11,0909	73,94	9,5962
37	394	8,0102	27,362	2,6138
38	574	7,4634	45,34	1,0235
39	385	9,7506	33,552	1,8560
40	1072	13,5933	78,436	4,2236
41	727	12,8377	66,237	1,7186
42	370	10,8324	28,753	1,8761
43	535	6,8766	32,979	2,5179
44	1004	8,0767	64,218	4,4079
45	223	12,4978	21,017	3,5805

Незалежними змінними в регресійній моделі будуть Classes (кількість класів проекту) та MBC (середня кількість методів у класах проекту), залежною змінною – KLOC (кількість строк коду проекту в тисячах).

Також в останньому стовпчику таблиці наведені розраховані значення SMD (квадрату відстані Махаланобіса d_i^2). Значення d_i^2 для кожної точки

i ($i = \overline{1, N}$) багатовимірних вихідних даних розраховується наступним чином [15]:

$$d_i^2 = (X_i - \bar{X})^T S_N^{-1} (X_i - \bar{X}), \quad (2.1)$$

де N – кількість елементів в вибірці,

X_i – i -а точка вибірки, $X_i = \{X_{1i}, X_{2i}, \dots, X_{mi}\}^T$,

m – кількість компонентів i – ї точки вибірки (для моделі з двома факторами $m = 3$),

$\bar{X}$ – вектор середніх значень компонентів вибірки,

S_N – вибіркова коваріаційна матриця [15]:

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

Вихідні дані нормально розподілені, тому що для всіх багатовимірних точок X_i SMD не перевищує квантіль розподілу χ^2 для рівня значимості 0,005 ($\chi^2=7,8147$).

Також підтверджує гаусівський багатовимірний розподіл вихідних даних значення багатовимірного ексцесу $\hat{\beta}_2$, для розрахунку якого використовується наступне співвідношення [17]:

$$\hat{\beta}_2 = \frac{1}{N} \sum_{i=1}^N \{(X_i - \bar{X})^T S_N^{-1} (X_i - \bar{X})\}^2. \quad (2.3)$$

Розраховане за (2.3) значення $\hat{\beta}_2$ дорівнює 14,6577. Відомо, що для нормально розподілених багатовимірних даних β_2 обчислюється за формулою: $\beta_2 = m(m + 2)$ (для моделі з двома факторами $m = 3$). В нашому випадку $\beta_2 = 3(3 + 2) = 15$. Тобто розраховане значення $\hat{\beta}_2$ менше β_2 для нормально розподілених даних.

Однією з передумов побудови моделі множинної регресії є перевірка незалежних факторів на мультиколінеарність. Суть мультиколінеарності полягає в тому, що два або більше факторів пов'язані між собою лінійною

залежністю або, тобто, мають високий ступінь кореляції. Основними наслідками мультиколінеарності є:

- зміщення оцінок параметрів моделі, які розраховуються за МНК. На основі цих оцінок неможливо зробити конкретні висновки про результати взаємозв'язку між показником і факторами;
- збільшення дисперсії та коваріації оцінок параметрів, обчислених за методом найменших квадратів;
- збільшення довірчого інтервалу (оскільки збільшується середній квадрат відхилення параметрів).

Єдиного способу визначення мультиколінеарності не існує. Один із методів визначення мультиколінеарності полягає в дослідженні коефіцієнтів інфляції дисперсії (VIFs), які використовуються для вимірювання сили кореляції між незалежними змінними моделі регресії. VIFs дорівнюють діагональним елементам оберненої кореляційної матриці незалежних змінних. В даному дослідженні VIFs будуть діагональними елементами оберненої кореляційної матриці 2-ох факторів. Отримали, що значення VIFs факторів Classes (кількість класів проекту) та MBC (середня кількість методів у класах проекту) не перевищують 5, що означає відсутність мультиколінеарності [18].

Для даних із таблиці 2.1 можна будувати множинну лінійну регресійну модель. Але нелінійна регресійна модель, зазвичай, має кращу якість та в більшості випадків вихідні дані не нормально розподілені. Тому побудуємо нелінійну регресійну модель, використавши метод нормалізуючих перетворень, основними етапами якого є:

- Нормалізація вихідних даних. В якості нормалізуючого перетворення обрано десятковий логарифм, так як це перетворення зазвичай використовується при побудові нелінійних регресійних моделей.

- Вилучення викидів. Для вилучення викидів з нормалізованих даних розраховано квадрат відстані Махаланобіса SMD для кожної багатовимірної точки i ($i = \overline{1, N}$) за формулою [15]:

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

де N – кількість багатовимірних точок,

Z_i – i -а нормалізована точка, $Z_i = \{Z_{1i}, Z_{2i}, \dots, Z_{mi}\}^T$,

m – кількість компонентів i – i точки (для моделі з двома факторами $m = 3$),

$\bar{Z}$ – вектор середніх нормалізованих значень компонентів,

S_N – коваріаційна матриця [15]:

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

Для визначення викидів в багатовимірних даних можемо порівнювати SMD з квантілем розподілу χ^2 з рівнем значимості α ($\chi_{m,\alpha}^2$). Та точка даних, для якої SMD перевищує $\chi_{m,\alpha}^2$, вважається викидом.

Також викиди можна визначати за допомогою тестової статистики T_{S_i} , яка для кожного SMD розраховується за формулою [19]:

$$T_{S_i} = (N - m) N d_i^2 / ((N^2 - 1) m). \quad (2.5)$$

Та точка даних, для якої статистика T_{S_i} більша, ніж $F_{m,N-m,\alpha}$ (квантіль F -розподілу з рівнем значимості α), вважається викидом.

Порівнявши d_i^2 з $\chi_{m,\alpha}^2$ та T_{S_i} з $F_{m,N-m,\alpha}$ на рівні значимості 0,005 для кожної з 45 багатовимірних точок, знайшли 1 викид по d_i^2 для 21 точки ($d_i^2 = 14,6411$; $\chi_{m,\alpha}^2 = 12,8382$). Після видалення викиду та перерахунку d_i^2 та T_{S_i} була відкинута 22 точка по невиконанню умови по d_i^2 та T_{S_i} . Залишилося 43 багатовимірні точки, для яких знову розраховано d_i^2 та T_{S_i} . На даній ітерації викиди відсутні.

2.2 Розробка двохфакторної нелінійної регресійної моделі для прогнозування розміру e-commerce додатків на Java

Для побудови двохфакторної нелінійної регресійної моделі для прогнозування розміру e-commerce додатків на Java методом нормалізуючих перетворень, виконано нормалізацію емпіричних даних з допомогою нормалізуючого перетворення на основі десяткового логарифму та перевірено багатовимірні нормалізовані дані на наявність викидів з допомогою квадрату відстані Махаланобіса та тестової статистики. Було знайдено та вилучено 2 викиди. Нормалізовані дані без викидів для побудови двохфакторної нелінійної регресійної моделі прогнозування розміру e-commerce додатків на Java, d_i^2 (2.4) та статистику T_{S_i} (2.5) наведено в таблиці 2.2.

Таблиця 2.2 – Нормалізовані дані для побудови двохфакторної нелінійної регресійної моделі прогнозування розміру e-commerce додатків на Java, d_i^2 та статистика T_{S_i}

№	Z_{X_1}	Z_{X_2}	Z_Y	d_i^2	T_{S_i}
1	2	3	4	5	6
1	2,9708	1,2063	1,8840	2,2794	0,7072
2	2,8745	0,9244	1,7593	1,6907	0,5245
3	2,9025	1,0718	1,7109	1,7119	0,5311
4	2,9074	0,9695	1,7394	1,3004	0,4034
5	2,8376	1,0196	1,6803	0,5993	0,1859
6	2,8082	1,4695	1,7045	6,0470	1,8761
7	2,7664	0,9276	1,5857	1,5206	0,4718
8	2,7825	1,0068	1,6298	0,5369	0,1666
9	2,7716	1,0405	1,6622	0,1458	0,0452
10	2,6875	1,4653	1,5547	7,5920	2,3554
11	2,7627	1,4578	1,5839	9,5931	2,9762
12	2,5647	0,8974	1,3562	5,3512	1,6602
13	2,7084	1,2130	1,5417	2,6108	0,8100
14	2,7093	1,2480	1,6539	0,7289	0,2261

Продовження табл. 2.2

1	2	3	4	5	6
15	2,9009	1,2922	1,8563	2,4686	0,7659
16	2,5551	1,3782	1,6013	4,3249	1,3418
17	2,7882	1,2291	1,7812	1,7084	0,5300
18	2,5988	0,9912	1,3805	4,8179	1,4947
19	2,6875	0,9810	1,5423	1,0442	0,3240
20	2,7686	1,4066	1,7929	4,0071	1,2432
23	2,9405	0,9548	1,8174	1,9071	0,5917
24	2,8927	0,9929	1,7174	1,1815	0,3666
25	2,7868	1,0250	1,7737	2,7955	0,8673
26	2,5024	1,2145	1,4260	2,8698	0,8903
27	2,9983	1,1575	1,8812	2,2868	0,7095
28	2,6866	1,1794	1,6315	0,4066	0,1262
29	2,3655	1,3894	1,4291	7,8443	2,4337
30	2,5623	0,9999	1,5977	5,8091	1,8022
31	2,9274	1,1295	1,9004	2,9376	0,9114
32	2,9365	0,9880	1,8230	1,6508	0,5121
33	2,7796	1,0324	1,7332	1,3053	0,4050
34	2,6776	1,3312	1,7852	6,7992	2,1094
35	2,7679	1,0115	1,6526	0,3009	0,0933
36	3,0788	1,0450	1,8689	4,2793	1,3276
37	2,5955	0,9036	1,4371	3,1366	0,9731
38	2,7589	0,8729	1,6565	2,3074	0,7159
39	2,5855	0,9890	1,5257	2,1524	0,6678
40	3,0302	1,1333	1,8945	2,7621	0,8569
41	2,8615	1,1085	1,8211	1,5537	0,4820
42	2,5682	1,0347	1,4587	1,8028	0,5593
43	2,7284	0,8374	1,5182	3,4222	1,0617

На наступному кроці побудови двохфакторної нелінійної регресійної моделі прогнозування розміру e-commerce додатків на Java буде розроблятися лінійна регресійна модель для 43 нормалізованих багатовимірних даних без викидів за формулою (1.7). При двох незалежних змінних формула прийме вигляд:

$$Z_Y = \hat{Z}_Y + \varepsilon = b_0 + b_1 Z_1 + b_2 Z_2 + \varepsilon, \quad (2.6)$$

де $Z_Y = \lg Y$; $Z_1 = \lg X_1$; $Z_2 = \lg X_2$; ε – випадкова величина.

Параметри моделі b_0, b_1, b_2 , які визначені за методом найменших квадратів, відповідно дорівнюють: $b_0 = -0,9398$, $b_1 = 0,8593$ та $b_2 = 0,2075$. Таким чином лінійна регресійна модель (2.6) для прогнозування розміру e-commerce додатків на Java, набуде вигляду:

$$Z_Y = \hat{Z}_Y + \varepsilon = -0,9398 + 0,8593Z_1 + 0,2075Z_2 + \varepsilon. \quad (2.7)$$

Для підтвердження правомірності використання лінійної регресійної моделі необхідно перевірити нормальність розподілу залишків ε із (2.7). Тобто необхідно перевірити гіпотезу H_0 , що полягає в тому, що ε розподілена за нормальним законом. Для великих вибірок з $N > 30$ для перевірити нульової гіпотези про нормальний розподіл випадкової величини можна використати критерій χ^2 Пірсона. Критерій χ^2 та інші статистичні критерії згоди використовуються для відкидання чи підтвердження гіпотези про характер розподілу емпіричних даних.

Сформувавши статистичну гіпотезу H_0 та вибравши статистичний критерій згоди Пірсона (χ^2) переходимо до розрахунку випадкової величини χ^2 по формулі [15]:

$$\chi^2 = \sum_{j=1}^m \frac{(n_j - Np_j)^2}{Np_j}, \quad (2.8)$$

де m – кількість підінтервалів розбиття інтервалу випадкової величини,

n_j – кількість значень випадкової величини в j -му підінтервалі,

p_j – ймовірність попадання випадкової величини в j -ий підінтервал.

Кількість підінтервалів m розбиття інтервалу випадкової величини може бути обчислена за формулою Брукса і Карузера $m = 5 \lg n$, або за формулою Стерджеса $m = 5 \log_2 n + 1 = 3,32 \lg n + 1$. В даній роботі використовувалася формула Стерджеса, за якою отримали 6 підінтервалів.

Ймовірність попадання випадкової величини в j -ий підінтервал визначалася за формулою [15]:

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

де x_{j-1}, x_j – нижня та верхня границі j -го підінтервалу,
 $f(x)$ – функція щільності ймовірності нормального розподілу.

Функція щільності ймовірності нормального розподілу знаходиться за формулою [15]:

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

де σ_x – вибіркове середньоквадратичне відхилення,
 m_x – математичне сподівання.

Наступним кроком перевірки нульової гіпотези є знаходження критичного значення $\chi^2_{кр}$ по таблиці χ^2 Пірсона. $\chi^2_{кр}$ визначається наступними параметрами: α – заданий рівень значимості (у нас 0,05, тобто з ймовірністю 0,95 можна гарантувати прийняття чи спростування гіпотези), ν – число степеней вільності, яке розраховується по формулі $\nu = m - k - 1$, де k – число параметрів розподілу (для нормального розподілу $k = 2$).

Якщо при порівнянні розрахованого χ^2 і знайденого $\chi^2_{кр}$ перше з них менше, нульова гіпотеза приймається, інакше – відхиляється. В даному випадку отримано: $\chi^2 = 0,57$, а $\chi^2_{кр} = 7,8147$. Тобто підтверджується нульова гіпотеза H_0 про розподіл випадкової величини ε за нормальним законом, що обґрунтовує правомірність використання лінійної регресійної моделі.

На наступному кроці побудови двохфакторної нелінійної регресійної моделі для прогнозування розміру e-commerce додатків на Java необхідно побудувати довірчий інтервал та інтервал прогнозування для лінійної регресії. Так як обґрунтована правомірність використання лінійної регресійної теорії за

допомогою критерію згоди Пірсона (χ^2), то визначити довірчий інтервал та інтервал прогнозування лінійної регресії (2.7) можна за відомими формулами (1.8), (1.9) та (1.10).

На наступному кроці, використавши зворотнє нормалізуюче перетворення, отримали двофакторну нелінійну регресійну модель для прогнозування розміру e-commerce додатків на Java:

$$Y = 10^{\varepsilon - 0,9398} X_1^{0,8593} X_2^{0,2075}. \quad (2.9)$$

Також через зворотнє нормалізуюче перетворення визначаємо границі довірчого інтервалу та інтервалу прогнозування для розробленої двофакторної нелінійної регресійної моделі для прогнозування розміру e-commerce додатків на Java (2.9).

При перевірці входження значень метрики KLOC з вихідних даних в інтервал прогнозування, виявили, що два значення (11 та 34) знаходяться за межами інтервалу прогнозування побудованої нелінійної регресійної моделі. Відповідні елементи вибірки були вилучені як викиди та знову повторений весь розрахунок для багатовимірних даних кількістю 41.

На цій ітерації викидів за Махаланобісом не знайдено, гіпотеза H_0 про нормальний закон розподілу залишків лінійної регресії ε підтвердилася, проте одне із значень метрики KLOC (30) не потрапило в інтервал прогнозування. Тому із вибірки був вилучений даний елемент, та здійснений перехід на наступну ітерацію. В вибірці залишилося 40 елементів.

На наступній ітерації знову одне із значень метрики KLOC (10) не потрапило в інтервал прогнозування. І знов з вибірки був вилучений відповідний 10 елемент. В вибірці залишилося 39 елементів.

Для 39 елементів, повторивши весь розрахунок, отримали:

– Лінійна регресійна модель для 39 нормалізованих даних має вигляд:

$$Z_Y = \hat{Z}_Y + \varepsilon = -1,1374 + 0,8966Z_1 + 0,2943Z_2 + \varepsilon.$$

– Гіпотеза H_0 про нормальний розподіл випадкової величини ε підтверджується ($\chi^2 = 0,9116$, $\chi_{кр}^2 = 7,8147$). Оцінка математичного сподівання залишків ε лінійної регресії для 39 нормалізованих даних дорівнює $-6,5E-16$, а оцінка середньоквадратичного відхилення дорівнює $0,0525$.

– Двохфакторна нелінійна регресійна модель для прогнозування розміру e-commerce додатків на Java для вибірки із 39 елементів має вигляд:

$$Y = 10^{\varepsilon-1,1374} X_1^{0,8966} X_2^{0,2943}.$$

– Всі значення метрики KLOC лежать в межах інтервалу прогнозування для побудованої моделі.

Також в роботі для вихідних даних побудована двофакторна лінійна регресійна модель для прогнозування розміру e-commerce додатків на Java:

$$Y = \hat{Y} + \varepsilon = 1,6081 + 0,0676X_1 + 0,3676X_2 + \varepsilon. \quad (2.10)$$

Якість побудованих моделей визначалася за наступними показниками: R^2 , $MMRE$ та $PRED(0,25)$ (1.11) - (1.15) [16, 17]. В таблиці 2.3 представлено розраховані значення R^2 , $MMRE$ та $PRED(0,25)$ лінійної та нелінійної двофакторних регресійних моделей для прогнозування розміру e-commerce додатків на Java.

Таблиця 2.3 – R^2 , $MMRE$ та $PRED(0,25)$ лінійної та нелінійної двофакторних регресійних моделей для прогнозування розміру e-commerce додатків на Java

Показники Модель	R^2	$MMRE$	$PRED(0,25)$
Лінійна двофакторна	0,8194	0,1281	0,8837
Нелінійна двофакторна	0,8743	0,0975	0,9231

Порівнюючи якість лінійної та нелінійної моделей, бачимо, що обидві моделі мають задовільні показники якості, а саме: R^2 та $PRED(0,25)$ більші ніж 0,75, а $MMRE$ менше 0,25. Але по всіх показникам нелінійна двохфакторна регресійна модель для прогнозування розміру мікросервісних e-commerce додатків на Java є кращою, ніж лінійна двохфакторна регресійна модель.

2.3 Висновки до розділу 2

В розділі була побудована двохфакторна нелінійна регресійна модель для прогнозування розміру мікросервісних e-commerce додатків на Java з використанням методу нормалізуючих перетворень та вилучення викидів.

Для цього:

- Виконаний збір емпіричних даних для побудови двохфакторної нелінійної регресійної моделі прогнозування розміру e-commerce додатків на Java, що розроблені на базі мікросервісів.
- Перевірені потенційні фактори на мультиколінеарність.
- Перевірені емпіричні дані на відповідність гаусівському розподілу.
- Виконана нормалізація вихідних даних за допомогою нормалізуючого перетворення у вигляді десяткового логарифму.
- Перевірені нормалізовані дані на викиди, використовуючи квадрат відстані Махаланобіса.
- Побудована двохфакторна лінійна регресійна модель для нормалізованих даних з обґрунтуванням правомірності її побудови;
- Побудовані довірчі інтервали та інтервали прогнозування для лінійної регресійної моделі.
- Побудовані, з використанням зворотнього перетворення, двохфакторна нелінійна регресійна модель прогнозування розміру e-commerce додатків на Java, що розробляються на базі мікросервісів, її довірчі інтервали та інтервали прогнозування, забезпечивши при цьому належність значень метрики KLOC інтервалам прогнозування.

- Визначена якість отриманої двохфакторної нелінійної регресійної моделі;
- Побудована двохфакторна лінійна регресійна модель для вихідних даних та визначена її якість;
- Виконане порівняння якості розроблених моделей.
- Виявлено, що двохфакторна нелінійна регресійна модель в порівнянні з двохфакторною лінійною регресійною моделлю має кращі показники якості.

3 ПРОЕКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПРОГНОЗУВАННЯ РОЗМІРУ МІКРОСЕРВІСНИХ E-COMMERCE ДОДАТКІВ НА JAVA

3.1 Постановка задачі на розробку ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java

В результаті аналізу публікацій за темою дослідження виявлено, що для прогнозування розміру програмних проектів використовуються, в тому числі, і регресійні моделі. Також виявлено, що якість регресійних моделей найкраща для даних, які схожі на ті, що використовувалися для побудови моделі. Для мікросервісних e-commerce додатків на Java не знайдено регресійної моделі для прогнозування розміру. Тому в кваліфікаційній роботі розроблено двохфакторну нелінійну регресійну модель для прогнозування розміру e-commerce додатків на Java.

На наступному етапі необхідно розробити програмне забезпечення, яке реалізує побудовану модель та надасть можливість виконувати раннє прогнозування розміру мікросервісних e-commerce додатків на Java.

Розроблюване програмне забезпечення повинно надавати наступні функціональні можливості:

- додавання даних для прогнозування розміру мікросервісних e-commerce додатків на Java;
- оновлення даних для прогнозування розміру мікросервісних e-commerce додатків на Java;
- отримання оцінки розміру мікросервісних e-commerce додатків на Java;
- отримання меж інтервалу прогнозування розміру мікросервісних e-commerce додатків на Java;

– отримання меж довірчого інтервалу розміру мікросервісних e-commerce додатків на Java.

Вхідними даними програмного забезпечення будуть: назва проекту, кількість класів проекту, середня кількість методів у класах проекту.

Вихідними даними програмного забезпечення будуть: кількість строк коду проекту в тисячах, верхня та нижня межі довірчого інтервалу, верхня та нижня межі інтервалу прогнозування.

Детальна постановка задачі представлена в технічному завданні (додаток А).

3.2 Ескізний проект ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java

Ескізний проект на розробку програмного забезпечення є початковим етапом, що включає загальні концепції та ідеї, які визначають основні аспекти майбутнього ПЗ. Він ґрунтується на обстеженні предметної галузі та технічному завданні.

У разі застосування структурної методології проектування в ескізному проекті зазвичай наводяться такі види інформації: контекстна діаграма, концептуальна модель даних, діаграма переходів станів та проект графічного інтерфейсу, які стануть основою для розробки технічного проекту [20].

3.2.1 Контекстна діаграма ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java

Контекстна діаграма ПЗ – це графічне зображення, яке показує, як програмне забезпечення взаємодіє із зовнішнім середовищем. Вона використовується для високорівневого представлення ПЗ та зв'язків із зовнішніми елементами (акторами, системами, пристроями).

Контекстна діаграма є частиною діаграм потоків даних (DFD, Data Flow Diagram) та відповідає 0-му рівню деталізації. Основна мета цієї діаграми – показати ПЗ як чорний ящик із входними й вихідними потоками даних без деталізації внутрішньої структури

Контекстна діаграма ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java зображена на рисунку 3.1.

Вхідними потоками даної діаграми є: запит на додавання даних для прогнозування розміру, запит на оновлення даних для прогнозування розміру, запит на прогнозування розміру. Вихідними потоками даної діаграми є: результат додавання даних для прогнозування розміру, результат оновлення даних для прогнозування розміру, результат прогнозування розміру. В якості зовнішньої сутності виступає користувач. Контекстним процесом є процес «Прогнозування розміру мікросервісних e-commerce додатків на Java».



Рисунок 3.1 – Контекстна діаграма ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java

Потоки даних контекстної діаграми та їх складові ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java наведені в словнику даних, представленому в таблиці 3.1. Контекстна діаграма містить два складові потоки, які в подальшому будуть декомпозовані.

Таблиця 3.1 – Словник даних контекстної діаграми ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java

Назва потоку	Складові потоку
запит на додавання даних для прогнозування розміру	назва проекту (Name), кількість класів проекту (Classes), Avg методів на клас проекту (MBC)
запит на оновлення даних для прогнозування розміру	назва проекту (Name), кількість класів проекту (Classes), Avg методів на клас проекту (MBC)
запит на прогнозування розміру	запит на оцінку розміру (запит) запит на межі інтервалу прогнозування (запит) запит на межі довірчого інтервалу (запит)
результат додавання даних для прогнозування розміру	назва проекту (Name), кількість класів проекту (Classes), Avg методів на клас проекту (MBC)
результат оновлення даних для прогнозування розміру	назва проекту (Name), кількість класів проекту (Classes), Avg методів на клас проекту (MBC)
результат прогнозування розміру	оцінка розміру (кількість тисяч строк) межі довірчого інтервалу (low довіри, up довіри) межі інтервалу прогнозування (low прогнозу, up прогнозу)

Після побудови контекстної діаграми необхідно розробити концептуальну модель даних.

3.2.2 Концептуальна модель даних ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java

Призначення концептуальної моделі даних полягає в тому, щоб на високому рівні відобразити, як структуровані дані в системі, і визначити основні сутності, їх атрибути, зв'язки між ними та загальні бізнес-вимоги до даних. Концептуальна модель даних ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java зображена на рисунку 3.2.

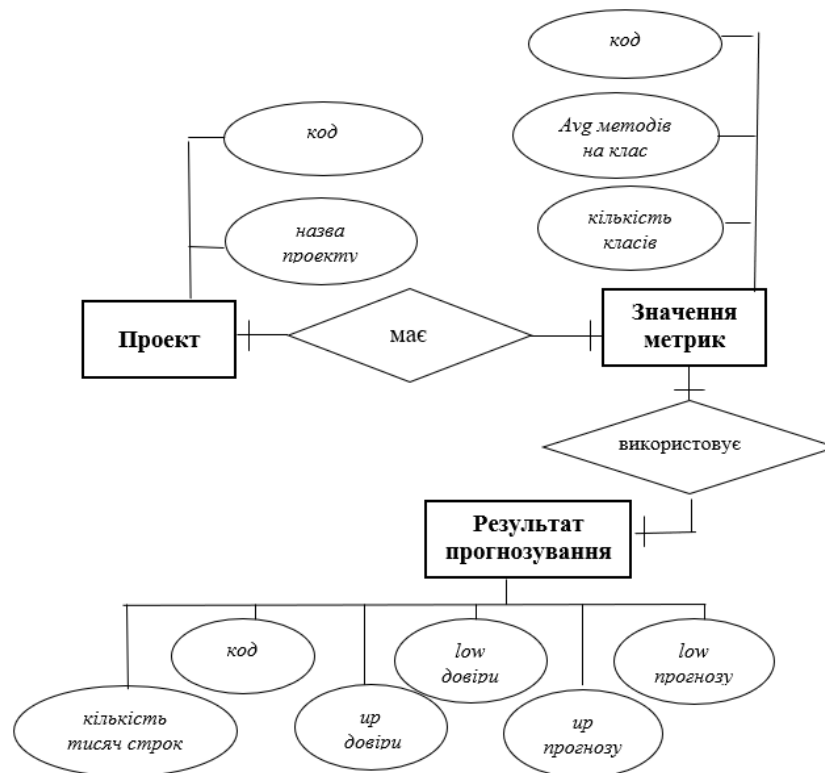


Рисунок 3.2 – Концептуальна модель даних ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java

Концептуальна модель даних містить наступні сутності: «Проект», «Результати прогнозування», «Значення метрик». Відношення між усіма сутностями один до одного.

Сутність «Значення метрик» має атрибути:

- код;
- Avg методів на клас;
- кількість класів.

Сутність «Проект» має атрибути:

- код;
- назва проекту

Сутність «Результат прогнозування» має атрибути:

- код;
- оцінка розміру;
- low довіри;
- up довіри;
- low прогнозу;
- up прогнозу.

3.2.3 Модель переходів станів ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java

Модель переходів станів програмного забезпечення використовується для опису логіки поведінки системи у різних станах залежно від певних подій чи умов. Вона надає формальний підхід до проектування, аналізу та тестування програмного забезпечення, забезпечуючи чітке уявлення про всі можливі стани, переходи між ними, а також події, що викликають ці переходи.

Модель переходів станів ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java представлена на рисунку 3.3.

Модель допомагає розробникам спроектувати структуру програми, логіку переходів і реакцій на події.

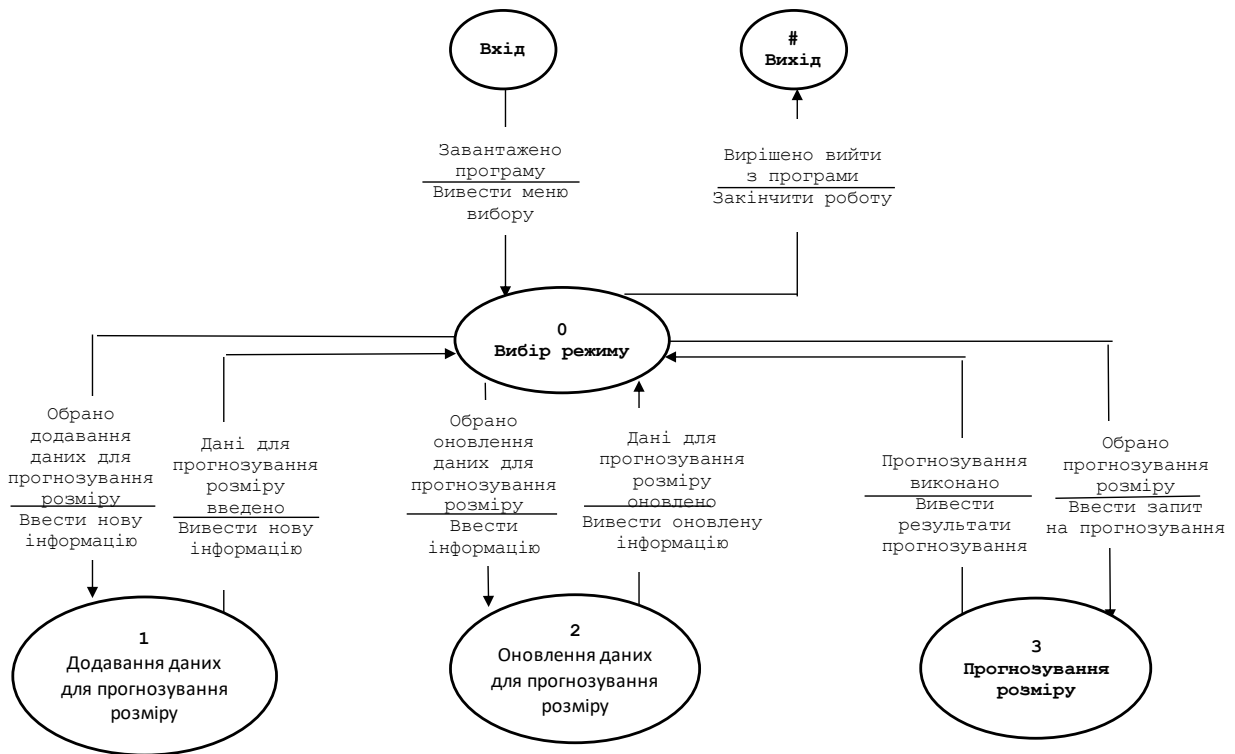
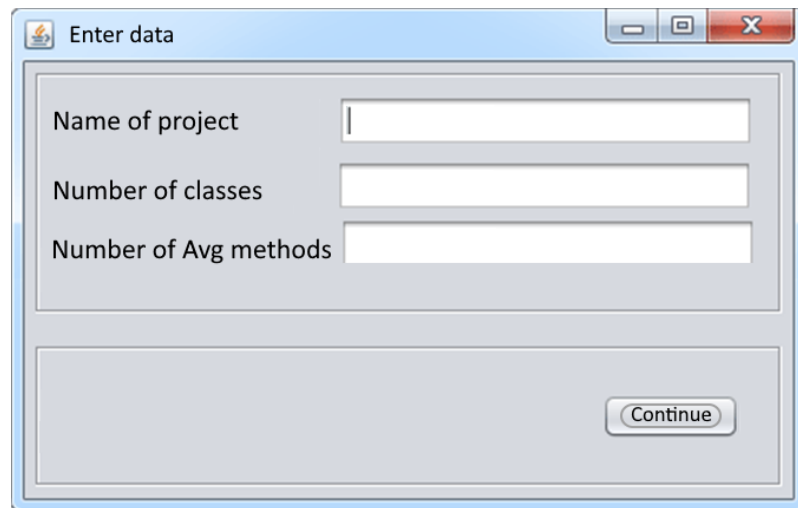


Рисунок 3.3 – Модель переходів станів ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java

3.2.4 Проект графічного інтерфейсу ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java

Розробка проекту графічного інтерфейсу ПЗ це невід'ємна частина процесу створення будь-якого програмного продукту. Вона починається з глибокого розуміння цільової аудиторії та її потреб. Основою для розробки є поєднання аналізу, проектування, тестування та постійного вдосконалення. Добре продуманий графічний інтерфейс допомагає досягти успішної взаємодії користувача із системою, підвищує конкурентоспроможність і робить продукт затребуваним. Використовуючи функціональні потоки предметної галузі побудовано проект графічного інтерфейсу ПЗ.

Проект форми графічного інтерфейсу для введення даних для прогнозування розміру мікросервісних e-commerce додатків на Java, представлений на рисунку 3.4.



Enter data

Name of project	
Number of classes	
Number of Avg methods	

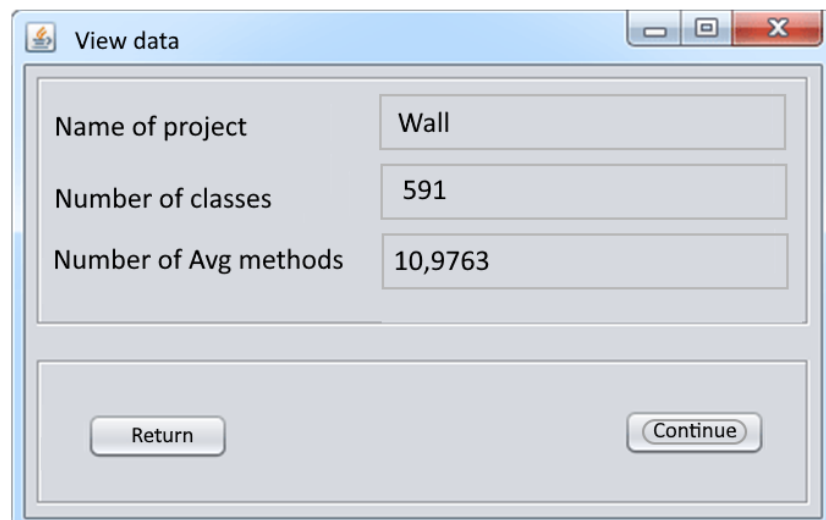
Continue

Рисунок 3.4 – Проект форми графічного інтерфейсу для введення даних для прогнозування розміру мікросервісних e-commerce додатків на Java

З допомогою кнопки «Continue» здійснюється перехід в наступне вікно.

На рисунку 3.5 представлено проект форми графічного інтерфейсу перегляду даних для прогнозування розміру мікросервісних e-commerce додатків на Java.

Вікно містить дві кнопки «Continue» та «Return». Призначення кнопки «Continue» – перехід в наступне вікно, «Return» – повернення до початкового.



View data

Name of project	Wall
Number of classes	591
Number of Avg methods	10,9763

Return Continue

Рисунок 3.5 – Проект форми графічного інтерфейсу для перегляду даних для прогнозування розміру мікросервісних e-commerce додатків на Java

На рисунку 3.6 представлений проект форми графічного інтерфейсу для представлення результатів прогнозування розміру мікросервісних e-commerce додатків на Java.

Project		
Wall		
Number of classes	Number of Avg methods	
591	10,9763	

KLOC		
KLOC: 45.0621		
	Low	Up
Predictional interval:	34.9116	58.1859
Confidence interval:	43.2208	46.9997

Return
Reset

Рисунок 3.6 – Проект форми графічного інтерфейсу для представлення результатів прогнозування розміру мікросервісних e-commerce додатків на Java

Кнопка «Return» призначена для повернення до початкового вікна, кнопка «Reset» – для завершення роботи програми.

3.3 Технічний проект ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java

Основою для розробки технічного проекту є ескізний проект. Його основне призначення – забезпечити розробників, тестувальників, замовників і інших учасників процесу розробки чітким планом для створення ПЗ. В технічному проекті наводиться деталізація компонентів системи, їхніх взаємозв'язків і залежностей [20].

В рамках технічного проекту розробляється ієрархія діаграм потоків даних, логічна модель даних та мініспецифікації підпроцесів.

3.3.1 Декомпозиція контекстного процесу ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java до рівня специфікацій програмних модулів

В технічному проекті розробляється ієрархія діаграм потоків даних, яка слугує графічним представленням потоків даних у системі і показує, як дані переміщуються між компонентами (процесами, базами даних, зовнішніми сутностями) і як вони обробляються. Ієрархія діаграм потоків даних використовується для поступового уточнення системи: від загального опису до детального розгляду окремих процесів.

Для кожного з простих процесів розробляються мініспецифікації, набір який утворює специфікацію системи, що становить основу для кодогенерації.

На рисунку 3.7 наведена декомпозиція контекстного процесу «Прогнозування розміру мікросервісних e-commerce додатків на Java».

Контекстний процес «Прогнозування розміру мікросервісних e-commerce додатків на Java» деталізується підпроцесами 1 «Додавання даних для прогнозування розміру», 2 «Оновлення даних для прогнозування розміру», 3 «Прогнозування розміру». Один із підпроцесів, а саме 3, є складним та потребує подальшої декомпозиції.

Декомпозиція процесу 3 «Прогнозування розміру» представлена на рисунку 3.8.

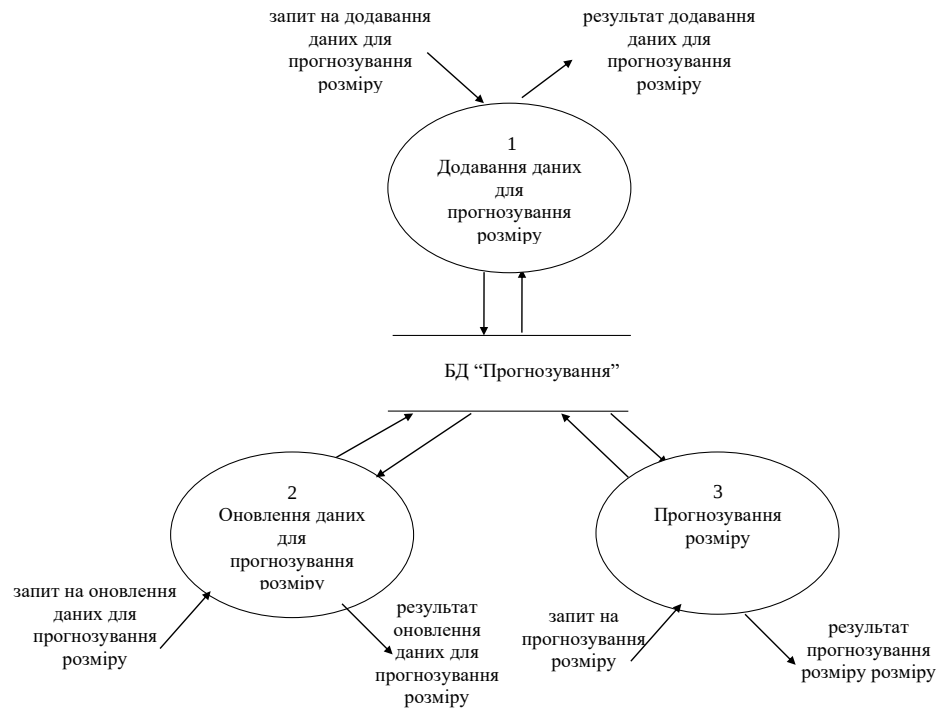


Рисунок 3.7 – Декомпозиція контекстного процесу «Прогнозування розміру мікросервісних e-commerce додатків на Java»

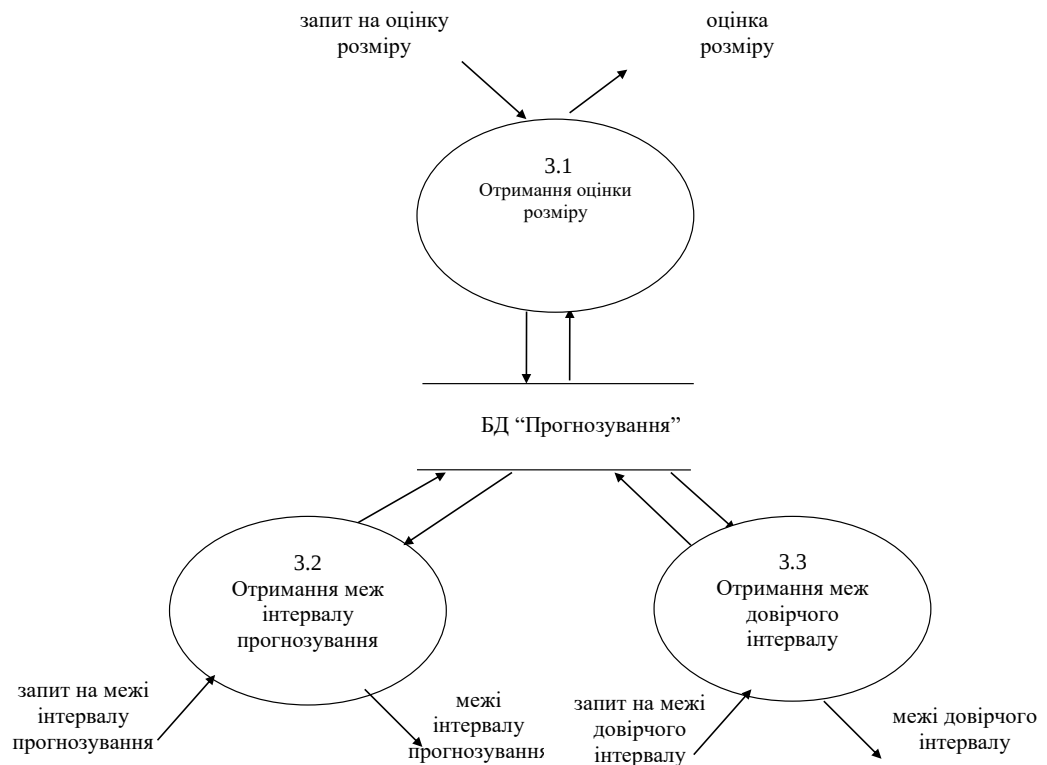


Рисунок 3.8 – Декомпозиція підпроцесу 3 «Прогнозування розміру»

Отримали всі прості підпроцеси. Ієрархія діаграм потоків даних побудована.

В таблиці 3.2 наведена відповідність потоків даних на діаграмах потоків даних всіх рівнів.

Таблиця 3.2 – Таблиця відповідності потоків даних на діаграмах потоків даних всіх рівнів

Потоки 0-го рівня	Потоки 1-го рівня	Потоки 2-го рівня	Атрибути потоків
1	2	3	4
запит на прогнозування розміру	запит на прогнозування розміру	запит на оцінку розміру	запит
		запит на межі інтервалу прогнозування	запит
		запит на межі довірчого інтервалу	запит
запит на додавання даних для прогнозування розміру	запит на додавання даних для прогнозування розміру	—	назва проекту (Name), кількість класів проекту (Classes), Avg методів на клас проекту (MBC)
запит на оновлення даних для прогнозування розміру	запит на оновлення даних для прогнозування розміру	—	назва проекту (Name), кількість класів проекту (Classes), Avg методів на клас проекту (MBC)

Продовження табл. 3.2

1	2	3	4
результат додавання даних для прогнозування розміру	результат додавання даних для прогнозування розміру	—	назва проекту (Name), кількість класів проекту (Classes), Avg методів на клас проекту (MBC)
результат прогнозування розміру	результат прогнозування розміру	оцінка розміру	кількість тисяч строк
		межі довірчого інтервалу	low довіри, up довіри
		межі інтервалу прогнозування	low прогнозу, up прогнозу
результат оновлення даних для прогнозування розміру	результат оновлення даних для прогнозування розміру	—	назва проекту (Name), кількість класів проекту (Classes), Avg методів на клас проекту (MBC)

Нижче наведені мініспецифікації для простих процесів.

Мініспецифікація підпроцесу 1 «Додавання даних для прогнозування розміру»

Вхідні дані: назва проекту (Name), кількість класів проекту (Classes), Avg методів на клас проекту (MBC)

Послідовність дій:

- вивести форму для введення інформації;
- ввести інформацію для прогнозування;

- вивести введену інформацію для прогнозування на екран.

Вихідні дані: назва проекту (Name), кількість класів проекту (Classes), Avg методів на клас проекту (MBC).

Мініспецифікація підпроцесу 2 «Оновлення даних для прогнозування розміру»

Вхідні дані: назва проекту (Name), кількість класів проекту (Classes), Avg методів на клас проекту (MBC).

Послідовність дій:

- вивести існуючу інформацію для прогнозування: назва проекту (Name), кількість класів проекту (Classes), Avg методів на клас проекту (MBC);
- ввести зміни в інформацію для прогнозування;
- вивести змінену інформацію для прогнозування на екран.

Вихідні дані: назва проекту (Name), кількість класів проекту (Classes), Avg методів на клас проекту (MBC).

Мініспецифікація підпроцесу 3.1 «Отримання оцінки розміру»

Вхідні дані: запит на оцінку розміру.

Послідовність дій:

- сформулювати запит до відповідних довідників;
- розрахувати прогнозне значення;
- вивести оцінку розміру на екран.

Вихідні дані: кількість тисяч строк коду.

Мініспецифікація підпроцесу 3.2 «Отримання меж інтервалу прогнозування»

Вхідні дані: запит.

Послідовність дій:

- сформулювати запит до відповідних довідників;
- розрахувати межі інтервалу прогнозування;
- вивести межі інтервалу прогнозування на екран.

Вихідні дані: low прогнозу, up прогнозу (ліва та права межа інтервалу прогнозування).

Мініспецифікація підпроцесу 3.3 «Отримання меж довірчого інтервалу»

Вхідні дані: запит.

Послідовність дій:

- сформулювати запит до відповідних довідників;
- розрахувати межі довірчого інтервалу;
- вивести межі довірчого інтервалу на екран.

Вихідні дані: low довіри, up довіри (ліва та права межа довірчого інтервалу).

3.3.2 Логічна модель даних ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java

На етапі побудови логічної моделі даних ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java, обрано реляційну модель даних.

Логічна модель даних ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java представлена на рисунку 3.9.

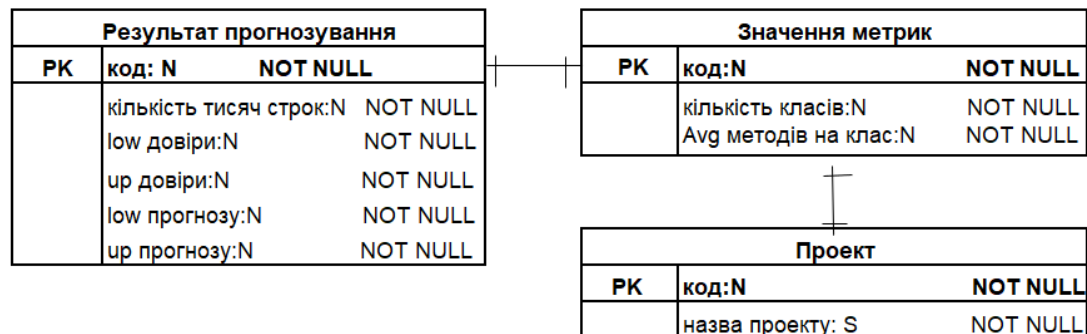


Рисунок 3.9 – Логічна модель даних ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java

Основними елементами реляційної моделі є відношення, атрибути, зв'язки між відношеннями, кортежі, первинні та зовнішні ключі. Зв'язки між відношеннями формуються через механізм зовнішніх ключів. Між відношеннями допустимі зв'язки один до одного та один до багатьох. При

побудові логічної моделі даних виконано нормалізацію. Модель даних приведена до третьої нормальної форми [21].

3.4 Робочий проект ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java

На етапі робочого проектування завершується розробка програмного забезпечення, тестування, випробування та розробка програмної документації [20].

3.4.1 Вибір засобів розробки ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java

В якості мови програмування обрана Java завдяки її численним перевагам, а саме: кросплатформеності, об'єктно-орієнтованості, безпеці, паралельному програмуванню, розвиненій екосистемі, автоматичному управлінню пам'яттю. Java застосовується для розробки серверних систем, мобільних додатків, ігор. Java характеризується довговічністю і підтримкою, легко інтегрується з іншими мовами.

Проте, як і інші мови, Java має ряд недоліків: низька продуктивність, Java-програми зазвичай повільніші, ніж аналогічні програми, написані на C++ або інших мовах низького рівня, можуть потребувати більше пам'яті та обчислювальних потужностей, особливо у порівнянні з мовами, оптимізованими для низькорівневих задач. Java вимагає більше коду для виконання тих самих задач, які в інших мовах (наприклад, Python чи Kotlin) можна реалізувати з меншими зусиллями. Є проблеми з оновленнями. Не підходить для систем реального часу. Існує конкуренція з новішими мовами. Так мови, як-наприклад, Kotlin, Scala або навіть Python, пропонують сучасніші підходи до програмування, через що Java може здаватися застарілою.

Java залишається універсальною мовою з багатою екосистемою та підтримкою, але її використання залежить від специфіки проекту. Вона

ідеально підходить для великих корпоративних систем і мобільних додатків, але може бути менш ефективною для проєктів, що вимагають максимальної продуктивності чи мінімального використання ресурсів.

У якості СКБД була обрана MySQL – одна з найпопулярніших реляційних систем керування базами даних, яка використовується для зберігання, організації та доступу до даних. MySQL має численні переваги, як наприклад, простота у використанні, висока швидкість та продуктивність, існує безкоштовна версія, безпека, велика спільнота розробників, активна розробка та підтримка, та інші. Також MySQL має і недоліки, проте їх значно менше, чим переваг. Тому, обираючи СКБД, зупинились на MySQL.

3.4.2 Кодування та тестування ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java

В якості мови програмування обрана Java завдяки її численным перевагам. У якості СКБД була обрана MySQL – одна з найпопулярніших реляційних систем керування базами даних. Програма та методика випробувань представлена в додатку Б, Опис програми знаходиться в додатку В, інструкція користувача – в додатку Г, текст програмного забезпечення – в додатку Д.

При тестуванні програмного забезпечення використовувався один із методів функціонального тестування, а саме метод еквівалентного розбиття. Тестування методом розбиття на класи еквівалентності – це техніка тестування ПЗ, яка дозволяє зменшити кількість тестових випадків, одночасно забезпечуючи адекватне покриття функціональності. Основна ідея методу полягає в тому, щоб поділити всі можливі вхідні дані на класи еквівалентності, де кожен клас містить значення, які, ймовірно, призведуть до однакового результату при тестуванні.

Переваги методу:

- економія часу і ресурсів;
- зменшення кількості тестів;

- виявлення помилок.

Таким чином, метод розбиття на класи еквівалентності є важливим інструментом у тестуванні програмного забезпечення, який дозволяє забезпечити ефективність і скоротити час тестування, при цьому гарантуючи перевірку основних сценаріїв.

Розглянемо тестування функції «Додавання даних для прогнозування розміру».

Мініспецифікація функції «Додавання даних для прогнозування розміру» наведена нижче.

Вхідні дані: назва проекту (Name), кількість класів проекту (Classes), Avg методів на клас проекту (MBC)

Послідовність дій:

- вивести форму для введення інформації;
- ввести інформацію для прогнозування;
- вивести введену інформацію для прогнозування на екран.

Вихідні дані: назва проекту (Name), кількість класів проекту (Classes), Avg методів на клас проекту (MBC).

В таблиці 3.3 представлені класи еквівалентності зазначеної функції.

Таблиця 3.3 – Класи еквівалентності функції «Додавання даних для прогнозування розміру»

Вхідні умови	Вірні класи еквівалентності	Невірні класи еквівалентності
назва проекту (Name),	1 строка символів	–
кількість класів проекту (Classes)	2 ціле додатнє число	3 ціле не додатнє число
		4 не ціле число
		5 не число
Avg методів на клас проекту (MBC)	6 дійсне додатнє число	7 дійсне не додатнє число
		8 не число

Розглянемо тести по вірним класам еквівалентності. Для вірних класів еквівалентності, якщо можливо, обираємо один тест на всі вірні класи. Тести по вірним класам еквівалентності знаходяться в таблиці 3.4.

Таблиця 3.4 – Тести по вірним класам

Тест	Клас	Значення	Результат
1	1	е-commerce додаток 1	Успішно зберегли інформацію для оцінювання розміру
	2	8	
	6	34,5	

Тести по невірним класам.

У таблиці 3.5 знаходяться тести по невірним класам. На кожен невірний клас потрібен свій тест.

Таблиця 3.5 – Тести по невірним класам

Тест	Клас	Значення	Результат
1	5	-101	Повідомлення про помилку
2	2	1,23	Повідомлення про помилку
3	7	ввар	Повідомлення про помилку
4	4	-5,42	Повідомлення про помилку
5	10	овар	Повідомлення про помилку

За результатами тестування програма працює вірно.

Аналогічно тестується функція «Оновлення даних для прогнозування розміру».

3.4.3 Випробування ПЗ для прогнозування розміру мікросервісних е-commerce додатків на Java

Випробування буде проведене згідно програми та методики випробувань, яка наведена в додатку Б. Під час випробування перевіряється

відповідності реальної поведінки програмного забезпечення його очікуваній поведінці на вибраному наборі тестів.

При випробуваннях перевірили всі функції:

- Завантажили програмне програмне «Size_prediction».
- В вікні «Enter data», що з'явилося, в поле для введення «Name of project» ввели назву проекту, в поле «Number of classes» – кількість класів у проекті, в поле «Number of Avg method» – середню кількість методів на клас та натиснули «Continue».
- В наступному вікні «View data» з'явилися введені дані для перегляду та дві кнопки «Return» та «Continue».
- При натисканні на кнопку «Return» повернулися в попереднє вікно для корегування даних.
- При натисканні на кнопку «Continue» перейшли в вікно «View prediction result» з вхідними даними «Name of project», «Number of classes», «Number of Avg method» та результатами прогнозування «KLOC» (оцінка розміру), «Predictional interval» (межі інтервалу прогнозування), «Confidence interval» (межі довірчого інтервалу). В вікні є дві кнопки «Return» та «Reset». При натисканні на кнопку «Return» повернулися в перше вікно для введення даних. При натисканні на кнопку «Reset» завершили роботу з програмою.

В результаті проведених випробувань впевнилися, що ПЗ працює відповідно до вимог, які були висунуті в технічному завданні.

3.5 Висновки до розділу 3

У розділі представлено розробку проекту ПЗ, який реалізує побудовану двохфакторну нелінійну регресійну модель для прогнозування розміру мікросервісних e-commerce додатків на Java.

При розробці проекту ПЗ виконано наступні роботи:

- Розроблено ескізний проект, який містить контекстну діаграму ПЗ, інтерфейс користувача, модель переходів станів, концептуальну модель даних.
- Розроблено технічний проект, під час роботи над яким побудовано ієрархію діаграм потоків даних, розроблено специфікацію ПЗ, розроблено логічну модель даних.
- Розроблено робочий проект, в рамках якого здійснений вибір засобів розробки програмного забезпечення, виконано кодування, тестування та випробування ПЗ.

4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ РОЗРОБКИ ТА ВПРОВАДЖЕННЯ ПЗ ДЛЯ ПРОГНОЗУВАННЯ РОЗМІРУ МІКРОСЕРВІСНИХ E-COMMERCE ДОДАТКІВ НА JAVA

4.1 Вступ

Дана робота присвячена розробці програми для прогнозування розміру мікросервісних e-commerce додатків на Java. Її можна використовувати для прогнозування обсягу та вартості робіт в комп'ютерних фірмах.

Ефективність програмних систем визначається ступенем їх позитивного впливу на підприємство, що управляється, або процес в результаті використання засобів обчислювальної техніки, програмного забезпечення, зміни інформаційних потоків та організаційної структури управління.

Створення програмного забезпечення вимагає одноразових витрат на розробку, придбання необхідних технічних засобів і поточних витрат на функціонування продукту. Економія від функціонування ПЗ визначається з урахуванням витрат на його експлуатацію. Відношення цієї економії до витрат на створення програмного забезпечення характеризує економічну ефективність капітальних вкладень. Економічні показники визначаються за діючими на момент розрахунку оптовими цінами, тарифами і ставками заробітної плати.

4.2 Розрахунок витрат на створення й експлуатацію програми для для прогнозування розміру мікросервісних e-commerce додатків на Java

Витрати на розробку системи складаються з витрат:

- на заробітну платню розробника;
- на амортизацію ЕОМ, на якій виконується розробка;
- на експлуатацію цієї ЕОМ;
- на засоби розробки;

- на матеріали і комплектуючі.

Розробка програмного забезпечення виконується програмістом, місячна заробітна плата якого складає 20000 грн.

Вартість сучасного ПК становить 40000 грн.

При вартості кіловат-години електроенергії 4.32 грн, розраховується вартість розробки програми.

Витрати на допоміжні матеріали наведені в табл. 4.1.

Таблиця 4.1 – Витрати на допоміжні матеріали

№	Пункти витрат	Сума, грн
1	Флеш-накопичувач даних	320
2	Папір для принтера	190
3	Картридж та тонер для принтера	240
	Разом	750

Вартість ПЗ знаходиться за формулою:

$$C_{\text{пр}} = (Z_{\text{зн}} + Z_{\text{сз}} + Z_{\text{зг}} + Z_{\text{е}}) * T + Z_{\text{м}} \quad (4.1)$$

де T – тривалість розробки, міс.

$Z_{\text{зн}}$ – основна і додаткова заробітна плата обслуговуючого персоналу, грн.

$Z_{\text{сз}}$ – відрахування на соціальні заходи (15% від основної і додаткової заробітної плати), грн.

$Z_{\text{зг}}$ – загальногосподарські витрати (10% від основної заробітної плати), грн.

$Z_{\text{е}}$ – витрати на електроенергію, грн.

$Z_{\text{м}}$ – витрати на основні і допоміжні матеріали, грн.

$Z_{\text{е}}$ при споживанні потужності 0.5 кВт, тривалості роботи на місяць, рівної $22 * 8 = 176$ годин, вартості кіловат-години електроенергії 4.32 грн становить:

$$З_е = 176 * 4.32 * 0.5 = 113.52 \text{ грн}$$

Витрати на розробку ПЗ наведені в табл. 4.2.

Таблиця 4.2 – Витрати на розробку програмного забезпечення

№	Найменування витрат	Одиниця	Кількість
1	Тривалість розробки (Т)	Міс.	3
2	Основна і додаткова заробітна плата ($З_{зп}$)	Грн.	20000.00
3	Відрахування на соціальні заходи ($З_{сз}$)	Грн.	3000.00
4	Загальногосподарські витрати ($З_{зг}$)	Грн.	2000.00
5	Витрати на допоміжні матеріали ($З_{м}$)	Грн.	750.00
6	Витрати на електроенергію ($З_е$)	Грн.	380,16

За формулою 4.1 розрахуємо вартість програми:

$$C_{пр} = (20000 + 3000 + 2000 + 380.16) * 3 + 750 = 76890.48 \text{ грн}$$

Амортизаційні відрахування складають 25% балансової вартості на рік:

$$A_{об} = 40000 * 0.25 = 10000 \text{ грн}$$

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

$$B_{м} = 40000 * 0.05 = 2000 \text{ грн}$$

Річний обсяг робіт ПК у годинах визначається в такий спосіб:

$$\Phi_{м} = 264.5 * T_3 \quad (4.2)$$

T_3 – це середнє місячне навантаження устаткування (близько 6 годин),
264.5 – середня кількість робочих днів на рік.

Таким чином, річний обсяг роботи ПК складає:

$$\Phi_{м} = 264.5 * 6 = 1587 \text{ годин}$$

Витрати на електроенергію Z_e при 1587 годинах роботи устаткування на рік становлять:

$$Z_e = 1587 * 4.32 * 0.5 = 3427.92 \text{ грн}$$

Експлуатаційні витрати на ПК на рік складуть:

$$Z_{зр} = 10000 + 2000 + 3427.92 = 15427.92 \text{ грн}$$

Отже, у перший рік витрати на створення й експлуатацію ПЗ складуть:

$$Z_p = 76890.48 + 15427.92 = 92318.40 \text{ грн}$$

4.3 Економічна ефективність розробки і впровадження ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java

Основним показником економічної ефективності програмного забезпечення є зменшення трудових витрат та часу прогнозування розміру ПЗ, а також ефективне розподілення трудових ресурсів на основі отриманих результатів.

Визначимо пряму економічну ефективність, ґрунтуючись на тому, що впровадження програмного продукту вивільняє 1.5 працівника (за експертною оцінкою фахівців підприємства).

Заробітна плата 1.5 працівника в рік складає:

$$Z = 20000 * 12 * 1.5 = 360000 \text{ грн}$$

Річний економічний ефект розраховується за формулою:

$$E_{\text{рік}} = \Delta C_n - E_n * k \quad (4.3)$$

де ΔC_n – вивільнені кошти після впровадження продукту (360000 грн) мінус експлуатаційні витрати (92318.40 грн) = 267681.6 грн;

E_n – коефіцієнт ефективності (дорівнює коефіцієнту амортизації 0.25);

k – одноразові витрати на впровадження продукту (76890.48 грн).

$$E_{\text{рік}} = 267681.6 - 0.25 * 76890.48 = 248458.89 \text{ грн}$$

Термін окупності ПЗ розраховується за формулою:

$$T = k / \Delta C_n = 76890.48 / 267681.6 = 0.29 \text{ року}$$

Отже, термін окупності ПЗ становить приблизно 4 місяці.

4.4 Висновки до розділу 4

У цьому розділі були здійснені розрахунки витрат, економічної ефективності та терміну окупності ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java. Виходячи з розрахунків, впровадження даного програмного забезпечення окупить себе протягом 4 місяців. Таким чином, його розробка є економічно обґрунтованою.

5 ОХОРОНА ПРАЦІ

5.1 Вступ

Охорона праці – це така система, що піклується про збереження життя та здоров'я працівників у ході їх трудової діяльності через правові, соціально-економічні, організаційно-технічні, санітарно-гігієнічні, лікувально-профілактичні, реабілітаційні та інші заходи.

Законодавство про працю містить норми і вимоги з техніки безпеки і виробничої санітарії, норми, що регулюють робочий час і час відпочинку, звільнення та переведення на іншу роботу, норми праці щодо жінок, молоді, гігієнічні норми і правила тощо.

Загальний нагляд за додержанням норм охорони праці покладено на прокуратуру, спеціальний – на професійні спілки. Контроль за безпекою праці здійснюють також державні й відомчі спеціалізовані інспекції.

Комп'ютерна техніка дуже широко застосовується у житті людини: вдома, на роботі та відпочинку, у різноманітних організаціях та виробництвах – інакше кажучи, комп'ютери міцно увійшли до повсякденного життя людей та масштаби цієї інтеграції з часом тільки збільшуються.

Недотримання вимог безпеки та експлуатації приводить до того, що з часом оператори комп'ютерної техніки починають відчувати перевтому та загальне погіршення стану здоров'я, що проявляється у погіршенні зору, головних болях, проблемах з хребтом, шиєю та кистями рук.

5.2 Аналіз шкідливих та небезпечних факторів на робочому місці програміста

Недостатнє освітлення

Освітлення – використання світлової енергії Сонця та штучних джерел світла для забезпечення можливості безперешкодного зорового сприйняття оточуючого середовища.

Світло є необхідною умовою життєдіяльності людини, воно потрібно для збереження здоров'я та високої продуктивності праці, що виявляється у якості проведених робіт та виробленої продукції.

Виступаючи подразником не тільки для органів зору, а і для всього організму, достатнє освітлення діє тонізуючи, покращує діяльність нервової системи та емоційний стан, настрій, впливає на формування добового ритму фізіологічних процесів організму людини.

Недостатнє освітлення приміщень та робочих поверхонь, навпаки, може потягнути за собою цілу купу негативних наслідків, котрі у результаті негативно позначаються на здоров'ї працівників та результатах їх праці. До впливів на здоров'я слід віднести такі прояви, як: втрата зору, головні болі, психологічні розлади тощо.

Згідно СНіП II-4-79, в приміщенні зі встановленими комп'ютерами, необхідно застосовувати систему комбінованого освітлення. У таких приміщеннях висуваються наступні вимоги щодо освітленості: при виконанні зорових робіт високої точності загальна освітленість повинна складати 300лк, а комбінована – 750лк; аналогічні вимоги при виконанні робіт середньої точності – 200 і 300лк відповідно. Крім того усе поле зору повинне бути освітлено максимально рівномірно – це основна гігієнічна вимога. Іншими словами, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими, тому що яскраве світло у районі периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до швидкого стомлення очей, що є однією з головних причин погіршення зору.

Параметри мікроклімату

Параметри мікроклімату можуть змінюватись у широких межах, в той час як необхідною умовою життєдіяльності людини є підтримка сталості

температури тіла завдяки терморегуляції, тобто здатності організму регулювати власну температуру через віддачу тепла в навколишнє середовище. Принцип нормування мікроклімату – створення оптимальних умов для теплообміну тіла людини з оточуючим середовищем, що зроблять знаходження людей у приміщенні комфортним та приємним.

Обчислювальна техніка є джерелом істотних тепловиділень, так, температура центрального процесора під час роботи може перевищувати 60°C, графічного ядра – більше ніж 50°C, тому для охолодження вузлів комп'ютера використовуються повітряні або водяні системи охолодження, що відводять тепло від елементів системи, розсіюючи його у навколишнє середовище. Такий вплив на мікроклімат приміщення може призвести до підвищення температури і зниження відносної вологості в приміщенні. У санітарних нормах СН-245-71 встановлені величини параметрів мікроклімату, що створюють комфортні умови для більшості працівників. Ці норми встановлюються залежно від пори року, характеру трудового процесу і характеру виробничого приміщення (дивись таблицю 5.1).

Таблиця 5.1 – Параметри мікроклімату для приміщень, де встановлені комп'ютери

Період року	Параметр мікроклімату	Величина
Холодний	Температура повітря в приміщенні	22...24°C
	Відносна вологість	40...60%
	Швидкість руху повітря	до 0,1м/с
Теплий	Температура повітря в приміщенні	23...25°C
	Відносна вологість	40...60%
	Швидкість руху повітря	0,1...0,2м/с

На кожну людину, що працює у приміщенні, об'єм цього приміщення повинен складати не менше ніж 19,5 м³ на особу з урахуванням максимального числа одночасно працюючих у зміну. Норми подачі свіжого повітря в приміщення, де розташовані комп'ютери, приведені в таблиці 5.2.

Таблиця 5.2 – Норми подачі свіжого повітря в приміщення, де розташовані комп'ютери

Характеристика приміщення	Об'ємна витрата свіжого повітря, що подається в приміщенні м ³ /на одну людину в годину
Обсяг до 20м ³ на людину	Не менше 30
20...40м ³ на людину	Не менше 20
Більше 40м ³ на людину	Природна вентиляція

Шум і випромінювання

Шум – це сукупність звуків різної частоти та інтенсивності, що виникають у результаті коливального руху частинок у пружних, тобто шумом можна назвати звуки котрі не не суть ніякого корисного інформаційного навантаження та своїм існуванням заважають працювати та концентруватися на певних задачах.

Шум погіршує умови праці, роблячи шкідливий вплив на організм людини: працюючи в умовах тривалого шумового впливу люди відчують дратівливість, головний біль, запаморочення, нервову втомленість, крім того, шум значно зменшує об'єм засвоєної інформації та підвищує кількість зроблених помилок під час виконання роботи порушуючи концентрацію.

Тривала дія інтенсивного шуму, рівень якого перевищує 80 дБА, на слух людини приводить до суттєвої втрати працеспроможності.

Рівень шуму на робочому місці не повинен перевищувати 50 дБА. Для зниження рівня шуму стіни і стеля приміщень, де встановлені комп'ютери, можуть бути облицьовані звукопоглинальними матеріалами, а самі робочі машини слід або ізолювати від працівників у інше приміщення, або використовувати якісні корпуси, що самі собі гарно ізолюють працюючі елементи системи від середовища, виступають бар'єром, та значно зменшують негативні впливи на людей.

Вважається, що як короткочасний, так і тривалий вплив усіх видів випромінювання від екрану монітора або системного блоку безпечно для

людини. Проте вичерпних даних щодо небезпеки дії випромінювання на працюючих з комп'ютерами не існує.

Максимальний рівень рентгенівського випромінювання на робочому місці оператора комп'ютера звичайно не перевищує 10мкбер/год, а інтенсивність ультрафіолетового і інфрачервоного випромінювань від екрану монітора лежить в межах 10...100мВт/м². Для зниження дії цих видів випромінювання рекомендується застосовувати якісні монітори та іншу периферійну техніку, а також дотримуватися регламенту періодів праці та відпочинку. Слід обмежити тривалість робочої діяльності біля екрана, не розміщувати дисплеї концентровано у робочій зоні, вимикати монітори, якщо за ним не працюють.

Ергономічні вимоги до робочого місця

Проектування робочих місць належить до важливих проблем ергономічного проектування у області обчислювальної техніки.

Робоче місце і взаємне розташування всіх його елементів повинне відповідати антропометричним, фізичним і психологічним вимогам. Велике значення має також характер роботи. Зокрема, при організації робочого місця співробітника повинні бути дотримані наступні основні умови: оптимальне розміщення обладнання, достатній робочий простір, що дозволяє здійснювати всі необхідні рухи і переміщення.

Ергономічними аспектами проектування робочих місць, зокрема, є: висота робочої поверхні, розміри простору для ніг, вимоги до розташування документів на робочому, характеристики робочого крісла, вимоги до поверхні робочого столу, можливість регулювання елементів робочого місця.

Головними елементами робочого місця співробітника є робочий стіл і крісло. Основним робочим положенням є положення сидячи.

Робоча поза сидячи викликає мінімальне стомлення співробітника(але водночас може привести до сколіозу через постійну, ледве помітну перевтому, що може виникати при порушенні режиму праці та відпочинку, та часто ігнорується співробітниками). Раціональна планування робочого місця

передбачає чіткий порядок і сталість розміщення предметів, засобів праці і документації. Те, що потрібно для виконання робіт частіше, має бути розташоване у зоні легкої досяжності робочого простору.

Для комфортної роботи оператора стіл повинен задовольняти наступним вимогам:

- така висота столу, щоб можна було зручно та вільно сидіти, спиратися на підлокітник якщо потрібно;
- вільне місце для ніг знизу;
- антиблікове покриття робочої поверхні столу;
- не менше трьох висувних ящиків.

Висота від підлоги до робочої поверхні повинна бути близько 680-760 мм. Висота поверхні, на яку встановлюється клавіатура – 650 мм.

Велике значення надається характеристикам робочого крісла. Так, висота сидіння над рівнем підлоги перебуває в межах 420-550 мм. Поверхня сидіння м'яка, передній край закруглений, а кут нахилу спинки – не фіксований, з можливістю регулювання.

Положення екрану визначається:

- відстанню від оператора до матриці (0,6 ... 0,7 м);
- кутом зчитування, напрямком погляду на 20° нижче горизонталі екрану.

Повинна також передбачатися можливість регулювання екрану:

- за висотою +/- 3 см;
- за нахилом від -10° до +20° по вертикалі;

Особливо важливою є поза користувача. При незручній робочій позі можуть з'явитися болі у м'язах, суглобах і сухожиллях, виникнути хронічні проблеми з поставою. Вимоги до робочої пози користувача наступні:

- голова не повинна бути нахилена більш ніж на 20°,
- плечі повинні бути розслаблені,
- лікті – під кутом 80° ... 100°,
- передпліччя і кисті рук – у горизонтальному положенні.

Причина неправильної пози користувачів обумовлена наступними чинниками: недостатній простір для ніг, не правильно відрегульоване крісло, монітор стоїть на неправильній відстані, стіл занадто великий або навпаки, робоча поверхня розташована занадто низько.

Вагоме значення для продуктивної і якісної роботи на комп'ютері мають розміри знаків, густина їх розміщення, контраст і співвідношення яскравості символів і фону екрану. Якщо відстань від очей до екрана дисплея становить 60 ... 80 см, то висота знака повинна бути не менше 3 мм, оптимальне співвідношення ширини і висоти знака складає 3:4, а відстань між знаками – 15 ... 20% їх висоти. Співвідношення яскравості фону екрану і символів – від 1:2 до 1:15.

Під час користування комп'ютером медики радять встановлювати монітор на відстані 50-60 см від очей. Фахівці також вважають, що верхня частина дисплея повинна бути на рівні очей або трохи нижче. Коли людина дивиться прямо перед собою, його очі відкриваються ширше, ніж коли він дивиться вниз. За рахунок цього площа огляду значно збільшується, викликаючи обезводнення очей. До того ж, якщо екран встановлений високо, а очі широко відкриті, порушується функція моргання: очі не закриваються повністю, чи не омиваються слізної рідиною, не отримують достатнього зволоження, що приводить до їх швидкої стомлюваності.

Створення сприятливих умов праці і правильне естетичне оформлення робочих місць на виробництві має велике значення як для полегшення праці, так і для підвищення його привабливості, якісно впливає на продуктивність.

5.3 Техніка безпеки при роботі на комп'ютері

Загальні положення:

1) При виконанні робіт на персональних комп'ютерах необхідно дотримуватись вимог загальної та даної інструкцій з охорони праці.

2) До самостійної роботи на комп'ютерах допускаються особи, які пройшли медичний огляд, навчання з професії, ввідний інструктаж з охорони праці і первинний інструктаж з охорони праці на робочому місці. У подальшому вони проходять повторні інструктажі з охорони праці на робочому місці один раз на півріччя.

3) Основним обладнанням робочого місця користувача ПК є: монітор, системний блок, клавіатура, друкуючий пристрій (принтер), зовнішні пристрої пам'яті, сканер, "миша", блок безперервного живлення, робочий стіл, стілець та інше; допоміжним – шафи, полиці, сейф та ін.

Робочі місця повинні бути розташовані на відстані не менше 1,5 м від стіни з вікнами, від інших стін – на відстані 1 м, між собою – на відстані не менше 1,5 м. Відносно вікон робочі місця доцільно розташувати таким чином, щоб природне світло падало на нього збоку, переважно зліва.

5) Монітори слід розташовувати таким чином, щоб запобігти потраплянню в очі прямого світла. Джерела освітлення слід розташовувати з обох боків екрану, паралельно напрямку зору.

Для запобігання світлових відблисків від екрана, клавіатури в напрямку очей користувача від освітлювачів загального призначення або сонячних променів необхідно застосовувати спеціальні фільтри, захисні козирки, на вікнах регульовані жалюзі.

Фільтри із металевої або нейлонової сітки використовувати не рекомендується тому, що сітка спотворює зображення внаслідок інтерференції світла. Найкраща якість зображення забезпечується скляними поляризаційними фільтрами – вони усувають практично усі відблиски і забезпечують чітке та контрастне зображення.

6) При роботі з текстовою інформацією (у режимі введення даних, редагування тексту і читання з екрану) найбільш фізіологічним є зображення чорних знаків на світлому (білому) фоні.

7) Розташовувати монітор на робочому місці слід так, щоб поверхня екрану знаходилась у центрі поля зору на відстані 400-700 мм від очей

користувача. Пропонується розташовувати елементи робочого місця таким чином, щоб підтримувалась однакова відстань від очей користувача до екрана, клавіатури, пулітра.

8) Робота комп'ютера супроводжується електромагнітним випромінюванням різних частот і рівнів, інтенсивність котрого зменшується пропорційно квадрату відстані до екрану. Оптимальною для працюючого за ПК є відстань 50 см від екрану монітору.

9) Рациональною робочою позою працюючого за ПК вважається таке розташування тіла, при якому ступні працівника розташовані горизонтально на підлозі або на підставці для ніг, стегна зорієнтовані в горизонтальній площині, верхні кінцівки рук – вертикально, кут ліктьового суглоба коливається в межах 70-90 град., зап'ястя зігнуті під кутом, не перевищує 20 град., нахил голови – у межах 15-20 град.

10) Для нейтралізації зарядів статичної електрики в приміщенні, де виконуються роботи на комп'ютерах, в тому числі на лазерних та струменевих принтерах, пропонується збільшувати вологість повітря за допомогою кімнатних зволожувачів. Не бажано носити одяг з синтетичних матеріалів.

Вимоги безпеки перед початком роботи

1) Перевірити надійність встановлення апаратури на робочому столі. Монітор не повинен стояти на краю столу. Повернути монітор треба так, щоб було зручно дивитись на екран – під прямим кутом (а не збоку) і трохи згори вниз, а екран повинен бути нахиленим під невеликим кутом (нижній край ближче до оператора).

2) Перевірити загальний стан апаратури, справність проводки електромережі, з'єднувальних шнурів, штепсельних вилок та розеток, заземлення захисного екрана.

4) Відрегулювати освітленість робочого місця.

5) Відрегулювати і зафіксувати висоту стільця (крісла).

6) Під'єднати до системного блока необхідну апаратуру (принтер, сканер і т.п.). Всі кабелі, що з'єднують системний блок з іншими пристроями слід вставляти і виймати тільки при вимкненому електричному живленні.

7) Увімкнути апаратуру ПК вмикачами на корпусах в такій послідовності : стабілізатор (або блок безперервного живлення), монітор, системний блок, принтер (якщо передбачається друкування).

8) Відрегулювати яскравість світіння екрану, фокусировку, контрастність. Не слід робити яскравість екрану занадто великою, тому що це втомлює очі.

Рекомендується:

- яскравість світіння екрану – не менше 100 кд/ кв.м (кандел на кв.м);
- відношення яскравості екрану до яскравості оточуючих його поверхонь в робочій зоні – не більше 3:1;
- мінімальний розмір точки світіння – не більше 0,4 мм для монохромного екрану і не більше 0,6 мм для кольорового;
- контрастність зображення знака – не менше 0,8.

9) При появі несправностей комп'ютерної техніки повідомити керівника.

Вимоги безпеки при закінченні роботи на ПК

1) Закінчити і зберегти в пам'яті ПК файли, які знаходились у роботі. Виконати всі дії для коректного завершення роботи в оперативній системі.

2) Вимкнути принтер та інші периферійні пристрої, вимкнути монітор і системний блок. Вимкнути стабілізатор (або блок безперервного живлення при його наявності). Штепсельні вилки витягнути з розеток. Накрити клавіатуру кришкою для попередження потрапляння в неї пилу.

3) Навести порядок на робочому місці.

4) Вимкнути освітлення та електроприлади.

Вимоги безпеки при аварійних ситуаціях

1) При раптовому припиненні подачі електричної енергії вимкнути всі пристрої ПК в такій послідовності: периферійні пристрої, монітор, системний

блок, стабілізатор (або блок безперервного живлення). Витягнути вилки з розеток.

2) При наявності ознак горіння (дим, запах горілого) необхідно вимкнути всі пристрої ПК, згідно з попереднім пунктом. Знайти місце загоряння і виконати всі можливі заходи для його ліквідації, попередивши терміново про це керівництво.

3) У випадку виникнення пожежі негайно попередити про це пожежну частину та керівництво, виконати усі можливі заходи щодо евакуації людей з приміщення і розпочати гасіння пожежі первинними засобами пожежогасіння.

6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

6.1 Аналіз існуючої проблеми необхідності охорони навколишнього середовища

Охорона навколишнього природного середовища, раціональне використання природних ресурсів, забезпечення екологічної безпеки життєдіяльності людини – невід’ємна умова сталого економічного та соціального розвитку України.

З цією метою Україна здійснює на своїй території екологічну політику, спрямовану на збереження безпечного для існування живої і неживої природи навколишнього середовища, захисту життя і здоров'я населення від негативного впливу, зумовленого забрудненням навколишнього природного середовища, досягнення гармонійної взаємодії суспільства і природи, охорону, раціональне використання і відтворення природних ресурсів.

Закон про охорону навколишнього середовища визначає правові, економічні та соціальні основи організації охорони навколишнього природного середовища в інтересах нинішнього і майбутніх поколінь.

Завданням законодавства про охорону навколишнього природного середовища є регулювання відносин у галузі охорони, використання і відтворення природних ресурсів, забезпечення екологічної безпеки, запобігання і ліквідації негативного впливу господарської та іншої діяльності на навколишнє природне середовище, збереження природних ресурсів, генетичного фонду живої природи, ландшафтів та інших природних комплексів, унікальних територій та природних об'єктів, пов'язаних з історико-культурною спадщиною.

Основними принципами охорони навколишнього природного середовища є:

а) пріоритетність вимог екологічної безпеки, обов'язковість додержання екологічних стандартів, нормативів та лімітів використання природних ресурсів при здійсненні господарської, управлінської та іншої діяльності;

б) гарантування екологічно безпечного середовища для життя і здоров'я людей;

в) запобіжний характер заходів щодо охорони навколишнього природного середовища;

г) екологізація матеріального виробництва на основі комплексності рішень у питаннях охорони навколишнього природного середовища, використання та відтворення відновлюваних природних ресурсів, широкого впровадження новітніх технологій;

д) збереження просторової та видової різноманітності і цілісності природних об'єктів і комплексів;

е) науково обгрунтоване узгодження екологічних, економічних та соціальних інтересів суспільства на основі поєднання міждисциплінарних знань екологічних, соціальних, природничих і технічних наук та прогнозування стану навколишнього природного середовища;

є) обов'язковість надання висновків державної екологічної експертизи;

ж) гласність і демократизм при прийнятті рішень, реалізація яких впливає на стан навколишнього природного середовища, формування у населення екологічного світогляду;

з) науково обгрунтоване нормування впливу господарської та іншої діяльності на навколишнє природне середовище;

и) безоплатність загального та платність спеціального використання природних ресурсів для господарської діяльності;

і) компенсація шкоди, заподіяної порушенням законодавства про охорону навколишнього природного середовища;

ї) вирішення питань охорони навколишнього природного середовища та використання природних ресурсів з урахуванням ступеня антропогенної змінності територій, сукупної дії факторів, що негативно впливають на екологічну обстановку;

й) поєднання заходів стимулювання і відповідальності у справі охорони навколишнього природного середовища;

к) вирішення проблем охорони навколишнього природного середовища на основі широкого міждержавного співробітництва;

л) встановлення екологічного податку, збору за спеціальне використання води, збору за спеціальне використання лісових ресурсів, плати за користування надрами відповідно до Податкового кодексу України.

6.2 Забруднення навколишнього середовища офісом комп'ютерної фірми

В процесі діяльності офісу комп'ютерної фірми утворюються різні тверді побутові відходи. До них можна віднести папір, картонні коробки, різні пакети та т.і., в цілому офісні відходи.

Проблема твердих побутових відходів характерна не тільки для цього закладу, з нею стикаються майже всі установи та заклади, які займаються подібною справою. Ця проблема є не тільки проблемою відокремлених підприємств, установ та закладів, але й всього міста та держави в цілому.

Проблема твердих побутових відходів (ТПВ) в Україні конкурує з показниками й аспектами цієї проблеми у світовому масштабі.

При сильній зволоженості та високій температурі повітря починається процес анаеробного розкладання ТПВ за рахунок розвитку анаеробних бактерій.

Несанкціоноване, стихійне складування ТПВ без урахування вимог і прийомів екологічної біотехнології викликає виділення шкідливих хімічних (сірководень, індол, скатол і т.п.) і біохімічних компонентів, які поширюючись

забруднюють ґрунтовий шар, попадають у ґрунтові води, а потім у відкриті водойми. Особливо всі ці несанкціоновані смітники ТПВ, що містять харчові відходи небезпечні в спекотну та суху пору року при $t > +25^{\circ}\text{C}$, коли дуже підсилюється розвиток всіх видів мікро- і макрофлори, мікро- і макрофауни, й що цілком природно, йде інтенсивна ферментація всіх харчових відходів і відходів природних полімерних матеріалів. При такій температурі $t > +25^{\circ}\text{C}$ створюються умови для інтенсивного розвитку й поширення найнебезпечніших інфекційних захворювань – холери й чуми. От чому й небезпечні стихійні смітники ТПВ не тільки для природного середовища, але й для людини.

У даний момент найпоширеніший спосіб знищення ТПВ – це полігони. Однак, цей простий спосіб супроводжують наступні проблеми:

- надмірно швидке переповнення існуючих полігонів через великий обсяг і малу щільність розташовуваних відходів. Без попереднього ущільнення середня щільність ТПВ становить 200-220 кг/м³, що досягає всього лише 450-500 кг/м³ після ущільнення з використанням сміттєвозів;

- негативні фактори для навколишнього середовища: зараження підземних вод сміттєвими продуктами, виділення неприємного запаху, розкид відходів вітром, мимовільне загоряння полігонів, безконтрольне утворення метану й неестетичний вид є тільки частиною проблем, що турбують екологів і визивають серйозні заперечення з боку місцевої влади;

- відсутність площ, придатних для розміщення полігонів на зручній відстані від великих міст. Розширення міст витісняє полігони на усе більш далекі відстані. Даний фактор у сполученні з ростом цін на землю збільшує вартість транспортування ТПВ;

- неможливість усунення полігонів. Незважаючи на використання найсучасніших технологій, наше суспільство завжди буде мати потребу в їхньому використанні для знищення не перероблених фракцій: золи, металобрухту, будівельного сміття.

6.3 Розробка заходів щодо зменшення забруднення

Збір відходів часто є найбільш дорогим компонентом усього процесу утилізації. Тому правильна організація збору відходів може заощадити значну кількість грошових витрат. Існуюча в Україні система збору ТПВ повинна залишатися стандартизованою з погляду економічності. У той же час додаткове планування необхідно для того, щоб вирішити нові проблеми (наприклад, відходи комерційних кіосків, на збір яких часто не вистачає ресурсів).

У густонаселених територіях нерідко доводиться транспортувати відходи на великі відстані. Рішенням у цьому випадку може бути станція тимчасового зберігання відходів, від якої сміття може вивозитися великими по вантажопідйомності машинами або по залізниці. При цьому треба відзначити, що станції проміжного зберігання являють собою об'єкти підвищеної екологічної небезпеки й можуть при неправильному розташуванні й експлуатації викликати не менше дорікань місцевих жителів і громадських організацій, ніж смітники.

Вторинна переробка. Досить багато компонентів ТПВ можуть бути перероблені в корисні продукти. Скло звичайно перероблюють шляхом здрібнювання й переплавлення (бажано, щоб вихідне скло було одного кольору). Скляний бій низької якості після здрібнювання використовується як наповнювач для будівельних матеріалів (наприклад, т.зв. «глассфальт»). У багатьох містах існують підприємства по відмиванню й повторному використанню скляного посуду. Така ж, безумовно, позитивна практика існує, наприклад, у Данії.

Сталеві й алюмінієві банки переплавляються з метою одержання відповідного металу. При цьому виплавка алюмінію з баночок для прохолодних напоїв вимагає тільки 5% від енергії, необхідної для виготовлення тієї ж кількості алюмінію з руди, і є одним з найбільш вигідних видів «ресайклінга».

Паперові відходи різного типу вже багато десятиків років застосовують поряд зі звичайною целюлозою для виготовлення пульпи – сировини для

паперу. Зі змішаних або низькоякісних паперових відходів можна виготовляти туалетний або обгортковий папір і картон. На жаль, в Україні тільки в невеликих масштабах присутня технологія виробництва високоякісного паперу з високоякісних відходів (обрізків друкарень, використаного паперу для ксероксів і лазерних принтерів і т.п.). Паперові відходи можуть також використовуватися в будівництві для виробництва теплоізоляційних матеріалів і в сільському господарстві – замість соломи на фермах.

Пластик. Переробка пластику досить дорогий та складний процес. З деяких видів пластику можна одержувати високоякісний пластик тих же властивостей, інші види пластику (наприклад ПВХ) після переробки можуть бути використані тільки як будівельні матеріали.

Компостування. Компостування – це технологія переробки відходів, заснована на їхньому природному біорозкладанні. Найбільш широко компостування застосовується для переробки відходів органічного, насамперед рослинного походження, таких як листи, вітки й скошена трава. Існують технології компостування харчових відходів та ТПВ.

У Україні компостування за допомогою компостних ям часто застосовується населенням в індивідуальних будинках або на садових ділянках. У той же час процес компостування може бути централізований і проводитися на спеціальних площадках. Існує кілька технологій компостування, що розрізняються за вартістю й складністю. Більше прості й дешеві технології вимагають більше місця й процес компостування займає більше часу.

Кінцевим продуктом компостування є компост, що може знайти різні застосування в міському й сільському господарстві.

Компостування, застосовуване в Україні на механізованих сміттєпереробних заводах, наприклад, у Львові, представляє собою процес зброджування в біореакторах усього обсягу ТПВ, а не тільки його органічної складової. Хоча характеристики кінцевого продукту можуть бути значно поліпшені шляхом добування з відходів металу, пластику й т.п., все ж таки він представляє собою досить небезпечний продукт і знаходить дуже обмежене

застосування (на Заході такий «компост» застосовують тільки для покриття смітників).

Таким чином, проблема побутових відходів є досить важливою для нашого суспільства. Існують розроблені методи для високотехнологічної утилізації сміття, але для того щоб вони були ефективними, треба, щоб кожна людина перейнялася відповідальністю за екологічний стан навколишнього середовища.

ВИСНОВКИ

Мета дослідження полягала в підвищенні достовірності прогнозування розміру e-commerce додатків на Java, що розробляються на базі мікросервісів, шляхом побудови двохфакторної нелінійної регресійної моделі і була повністю досягнута. Досягнення мети забезпечилося виконанням наступних завдань дослідження:

- аналіз наявних методів і моделей для прогнозування, на ранніх етапах роботи над проектом, розміру e-commerce додатків на Java, що розробляються на базі мікросервісів;

- збір емпіричних даних для побудови двохфакторної нелінійної регресійної моделі прогнозування розміру e-commerce додатків на Java, що розроблені на базі мікросервісів;

- перевірка потенційних факторів на мультиколінеарність;

- перевірка емпіричних даних на відповідність гаусівському розподілу;

- виконання нормалізації вихідних даних за допомогою нормалізуючого перетворення у вигляді десяткового логарифму;

- вилучення викидів в даних, використовуючи квадрат відстані Махаланобіса;

- побудова двохфакторної лінійної регресійної моделі з обґрунтуванням правомірності її побудови;

- побудова довірчих інтервалів та інтервалів прогнозування для лінійної регресійної моделі;

- побудова, з використанням зворотнього перетворення, двохфакторної нелінійної регресійної моделі прогнозування розміру e-commerce додатків на Java, що розробляються на базі мікросервісів, її довірчих інтервалів та інтервалів прогнозування, забезпечивши при цьому належність розміру проекту з емпіричних даних інтервалам прогнозування;

- оцінювання якості отриманої двохфакторної нелінійної регресійної моделі;

- виконання програмної реалізації двохфакторної нелінійної регресійної моделі прогнозування розміру e-commerce додатків на Java, що розробляються на базі мікросервісів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Kaczmarek J., Kucharski M. Size and effort estimation for applications written in Java. *Information and Software Technology*, 2004. Vol. 46, Issue 9, pp. 589-601.
2. K. Tan, Y. Zhao, H. Zhang. Estimating LOC for information systems from their conceptual data models. *Proceedings of the 28th International Conference on Software Engineering (ICSE '06)* (May 20-28, 2006). Shanghai, China, 2006, pp. 321-330.
3. K. Tan, Y. Zhao, H. Zhang. Conceptual data model-based software size estimation for information systems. *Transactions on Software Engineering and Methodology*. 2009. Vol. 19, Issue 2, pp. 1-37.
4. Prykhodko N.V., Prykhodko, S.B. The non-linear regression model to estimate the software size of open source java-based systems. *Радіоелектроніка, інформатика, управління*, 2018, 3(46), 158-166.
5. Макарова Л.М., Приходько Н.В., Кудін О.О. Побудова нелінійної регресійної моделі для оцінювання розміру веб-додатків, реалізованих мовою Java. *Вісник Херсонського національного технічного університету*, 2019, 2 (69), 145-153.
6. Приходько С.Б., Приходько Н.В. Смикодуб Т.Г. Чотирьохфакторна нелінійна регресійна модель для оцінювання розміру Java-застосунків з відкритим кодом. *Вчені записки ТНУ імені В.І. Вернадського. Серія: технічні науки*. Київ, 2020. Том 31(70) Ч. 1 № 2. С. 157-162.
7. Латанська Л.О., Карлович Д.Г. Множинна нелінійна регресія для оцінювання розміру мікросервісів на Java, які розробляються з використанням Spring фреймворку. *Інформаційні технології: моделі, алгоритми, системи: зб. тез IV всеукраїнської науково-практичної інтернет конференції, 30-31 жовтня 2023р.* 2023, С 18-19.
8. Montgomery D.C., Peck E.A., Vining G.G. Introduction to Linear Regression Analysis. 5th ed. Wiley, 2012, 672 p.

9. Stol K., Fitzgerald B. Researching software development: From prediction to understanding. *Journal of Systems and Software*, 2016, Vol. 117, pp. 76-91.
10. Пухалевич А.В., Алієв А.Ш. Передобробка даних при побудові регресійної моделі для прогнозування розміру e-commerce додатків на Java, розроблених на базі мікросервісів. *Матеріали VII Всеукраїнської науково-практичної інтернет-конференції студентів, аспірантів та молодих вчених за тематикою «Сучасні комп'ютерні системи та мережі в управлінні»:* збірка наукових праць (29 листопада 2024 року, м. Херсон, м. Хмельницький) / за ред. А.А. Григорової. – Херсон: Видавництво ФОП Вишемирський В. С., 2024.– С.198-199.
11. Fenton N., Bieman J. Software metrics: A Rigorous and Practical Approach, 3rd Edition. CRC Press, 2014, 638 p.
12. Новичков А., Шамрай А., Черников А. Метрики кода и их практическая реализация в Subversion и ClearCase. URL: <http://gdznet.com/userhelps/metriki-koda-i-ix-prakticheskaya-realizaciya-v-subversion-i-clearcase-chast-1-metriki.html> (дата звернення: 20.04.2024).
13. Boehm B.W. Software engineering economics. Prentice-Hall, 1981, 320 p.
14. Johnson Kim. Software cost estimation – Metrics and models. University of Calgary, 2001, 115 p.
15. 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.
16. Fenton N., Bieman J. Software Metrics: A Rigorous and Practical Approach, 3rd ed., CRC Press, 2014, 617 p.
17. Mardia K.V. Measures of multivariate skewness and kurtosis with applications. *Biometrika*. 1970. Vol. 57, Issue 3. P. 519-530.
18. Коэффициент инфляции дисперсии (VIF). URL: <http://surl.li/mfqhw> (дата звернення: 10.10.2024).

19. Латанська Л.О. Нелінійна регресійна модель для оцінювання розміру веб-застосунків, що створюються з використанням PHP фреймворку Symfony / Л.О. Латанська, Л.М. Макарова, А.В. Кольцов, Д.Х. Давлатова // «Вісник Хмельницького національного університету. Серія: Технічні науки». – Хмельницький: ХНУ, 2022. – № 6 (315). – С. 119-124.
20. Табунщик Г.В., Каплієнко Т.І., Петрова О.А. Проектування та моделювання програмного забезпечення сучасних інформаційних систем. Запоріжжя, Дике Поле, 2016. 250 с.
21. Гайна Г.А. Основи проектування баз даних. Навчальний посібник. Київ, КНУБА, 2005. 204 с.

ДОДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ ПЗ ДЛЯ ПРОГНОЗУВАННЯ РОЗМІРУ МІКРОСЕРВІСНИХ Е-COMMERCE ДОДАТКІВ НА JAVA

Вступ

Назва програми – ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java.

Програма надасть можливість користувачу виконати прогнозування розміру мікросервісних e-commerce додатків на Java на ранніх етапах розробки.

1 Підстави для розробки

Документом, на підставі якого ведеться розробка ПЗ, є наказ ректора НУК про затвердження тем кваліфікаційних робіт магістрів.

2 Призначення розробки

2.1 Функціональне призначення

Програмний продукт буде використовуватись для прогнозування розміру мікросервісних e-commerce додатків на Java і побудови довірчого інтервалу та інтервалу прогнозування.

2.2 Експлуатаційне призначення

Даний програмний продукт дозволить автоматизувати та скоротити час відповідних розрахунків.

3 Вимоги до програми

3.1 Вимоги до функціональних характеристик

3.1.1 Вимоги до складу функцій, що виконуються

Програмне забезпечення повинно мати наступний функціонал:

- додавання даних для прогнозування розміру мікросервісних e-commerce додатків на Java;
- оновлення даних для прогнозування розміру мікросервісних e-commerce додатків на Java;
- отримання оцінки розміру мікросервісних e-commerce додатків на Java;
- отримання меж інтервалу прогнозування розміру мікросервісних e-commerce додатків на Java;
- отримання меж довірчого інтервалу розміру мікросервісних e-commerce додатків на Java.

3.1.2 Вимоги до організації вхідних та вихідних даних

Вихідні дані мають зберігатися у таблицях реляційної бази даних. Внесення вхідних даних має виконуватися шляхом їх введення у спеціальних формах.

Результуючі дані (результат оцінювання розміру, визначений довірчий інтервал, визначений інтервал прогнозування) повинні записуватися у таблицю реляційної бази даних та виводитися в відповідних місцях в створеній програмою формі.

3.2 Вимоги до надійності

3.2.1 Вимоги до забезпечення надійності функціонування програми

Надійне функціонування програми має бути забезпечене надійним функціонуванням операційної системи.

3.2.2 Відмова виконання програмного продукту через некоректні дії користувача

Відмова програми можлива внаслідок некоректних дій користувача при взаємодії з операційною системою. Щоб уникнути виникнення відмов програми за вказаною вище причиною необхідно забезпечити стабільне функціонування операційної системи.

3.4 Умови експлуатації

Кліматичні умови експлуатації програмного продукту відповідають умовам експлуатації апаратної частини персонального комп'ютера.

3.5 Вимоги до інформаційної та програмної сумісності

Системні програмні засоби, що використовуються програмою, повинні бути представлені ліцензійною версією ОС Windows 7 та вище.

3.6 Вимоги до захисту інформації та програм

Вимоги до захисту інформації та програм відсутні.

3.7 Вимоги до маркування та упаковки

3.7.1 Вимоги до маркування

Вимоги до маркування відсутні.

3.7.2 Вимоги до упаковки

Вимоги до упаковки відсутні.

3.8 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання програми відсутні.

4 Вимоги до програмної документації

Склад програмної документації:

- технічне завдання;
- текст програми;
- опис програми;
- програма та методика випробувань;
- інструкція користувача.

5 Техніко-економічні показники

Економічна ефективність впровадження програмного забезпечення розрахована у розділі 4. Згідно з отриманими результатами, впровадження даного ПЗ є доцільним, а термін його окупності становить приблизно 4 місяці.

6 Стадії та етапи розробки програмного забезпечення

Стадії та етапи розробки ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java зазначені в таблиці А.1.

Таблиця А.1 – Стадії та етапи розробки програмного забезпечення

Стадії розробки	Етапи робіт	Терміни виконання
Технічне завдання	Обумовлення необхідності розробки	З 10.09.2024 до 13.09.2024
	Розробка та затвердження технічного завдання	З 14.09.2024 до 19.09.2024
Ескізний проект	Розробка ескізного проекту	З 20.09.2024 до 9.10.2024
	Затвердження ескізного проекту	З 10.10.2024 до 13.10.2024
Технічний проект	Розробка технічного проекту	З 14.10.2024 до 03.11.2024
	Затвердження технічного проекту	З 04.11.2024 до 8.11.2024
Робочий проект	Розробка ПЗ	З 9.11.2024 до 20.11.2024
	Розробка програмної документації	З 21.11.2024 до 02.12.2024
	Випробування програми	З 03.12.2024 до 08.12.2024

7 Порядок контролю та приймання

Контроль здійснюється замовником на кожній стадії розробки ПЗ. Приймання здійснюється замовником згідно програми та методики випробувань. Випробування проводиться в зазначений термін.

За результатами приймально-здавальних робіт складається акт приймання-здачі програми в експлуатацію, який підписується організаціями розробника та замовника.

ДОДАТОК Б

ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ ПЗ ДЛЯ ПРОГНОЗУВАННЯ РОЗМІРУ МІКРОСЕРВІСНИХ E-COMMERCE ДОДАТКІВ НА JAVA

1 Об'єкт випробувань

Об'єктом випробувань є ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java.

2 Мета випробувань

Метою випробувань є перевірка функціональних можливостей програмного забезпечення на відповідність вимогам технічного завдання. У випадку прийнятного рівня відповідності програмного забезпечення вимогам, відсутності виявлених помилок і грубих недоліків, програмне забезпечення приймається в експлуатацію.

3 Вимоги до програми

В ході проведення випробувань програмного продукту необхідно перевірити, що розроблене ПЗ реалізує наступні функції:

- додавання даних для прогнозування розміру мікросервісних e-commerce додатків на Java;
- оновлення даних для прогнозування розміру мікросервісних e-commerce додатків на Java;
- отримання оцінки розміру мікросервісних e-commerce додатків на Java;
- отримання меж інтервалу прогнозування розміру мікросервісних e-commerce додатків на Java;
- отримання меж довірчого інтервалу розміру мікросервісних e-commerce додатків на Java.

4 Вимоги до програмної документації

Склад програмної документації, що надається на випробування:

- технічне завдання;
- текст програми;
- програма і методика випробувань;
- опис програми;
- інструкція користувача.

5 Засоби і порядок випробувань

Технічні засоби випробувань:

- не менше 1 ГБ оперативної пам'яті;
- не менше 500 Мб вільного місця на жорсткому диску;
- клавіатура та миша.

Програмні засоби випробувань:

- ОС Windows 7 та вище.

Порядок проведення випробувань.

- виконуються інструкції до кожної функції;
- робиться аналіз отриманих результатів і встановлюється відповідність

програмного продукту вимогам і системним документам.

При виявленні помилок у роботі ПЗ складається їх перелік і обговорюється термін їх виправлення розробником. Після цього замовник проводить повторне тестування в повному обсязі.

6 Методи випробувань

Випробування будемо проводити одним із методів функціонального тестування, а саме методом розбиття на класи еквівалентності. Всі функції програми будуть проходити випробування. Розглянемо випробування програмного забезпечення:

- Завантажити програмне програмне «Size_prediction».

– В вікні «Enter data», що має з'явитися, в поле для введення «Name of project» ввести назву проекту, в поле «Number of classes» – кількість класів у проекті, в поле «Number of Avg method» – середню кількість методів на клас та натиснули «Continue».

– В наступному вікні «View data» повинні з'явитися введені дані для перегляду та дві кнопки «Return» та «Continue».

– При натисканні на кнопку «Return» маємо повернутися в попереднє вікно для корегування даних.

– При натисканні на кнопку «Continue» маємо перейти в вікно «View prediction result» з вхідними даними «Name of project», «Number of classes», «Number of Avg method» та результатами прогнозування «KLOC» (оцінка розміру), «Predictional interval» (межі інтервалу прогнозування) , «Confidence interval» (межі довірчого інтервалу). В вікні мають бути дві кнопки «Return» та «Reset». При натисканні на кнопку «Return» маємо повернутися в перше вікно для введення даних. При натисканні на кнопку «Reset» маємо завершити роботу з програмою.

ДОДАТОК В

ОПИС ПЗ ДЛЯ ПРОГНОЗУВАННЯ РОЗМІРУ МІКРОСЕРВІСНИХ E-COMMERCE ДОДАТКІВ НА JAVA

1 Загальні відомості

1.1 Позначення і найменування програми

Найменування: ПЗ для прогнозування розміру мікросервісних e-commerce додатків на Java.

1.2 Мови програмування

В якості мови програмування використовувалася Java.

2 Функціональне призначення

Класи вирішуваних завдань і призначення програми

Програмне забезпечення «Size_prediction» повинно мати наступний функціонал:

- додавання даних для прогнозування розміру мікросервісних e-commerce додатків на Java;
- оновлення даних для прогнозування розміру мікросервісних e-commerce додатків на Java;
- отримання оцінки розміру мікросервісних e-commerce додатків на Java;
- отримання меж інтервалу прогнозування розміру мікросервісних e-commerce додатків на Java;
- отримання меж довірчого інтервалу розміру мікросервісних e-commerce додатків на Java.

3 Використовувані технічні засоби

- не менше 1 ГБ оперативної пам'яті;
- не менше 500 Мб вільного місця на жорсткому диску;
- клавіатура та миша.

4 Виклик і завантаження

Запуск ПЗ «Size_prediction» в графічному режимі здійснюється двома способами:

- 1) за допомогою файлу запуску: `java -jar size_prediction.jar`
- 2) з використанням командного рядку: `java -jar size_prediction.jar`

5 Вихідні та результуючі дані

Вхідними даними програми є назва проекту, кількість класів проекту, середня кількість методів у класах проекту. Ці дані повинні вводитися в спеціальну форму в відповідні поля.

Вихідними даними програмного забезпечення будуть: кількість строк коду проекту в тисячах, верхня та нижня межі довірчого інтервалу, верхня та нижня межі інтервалу прогнозування.

.

ДОДАТОК Г

ІНСТРУКЦІЯ КОРИСТУВАЧА ПЗ ДЛЯ ПРОГНОЗУВАННЯ РОЗМІРУ МІКРОСЕРВІСНИХ E-COMMERCE ДОДАТКІВ НА JAVA

Запустити програмне забезпечення Size_prediction. В вікні «Enter data», що з'явилося, в поле для введення «Name of project» ввести назву проекту, в поле «Number of classes» – кількість класів у проекті, в поле «Number of Avg method» – середню кількість методів на клас (рисунок Г.1).

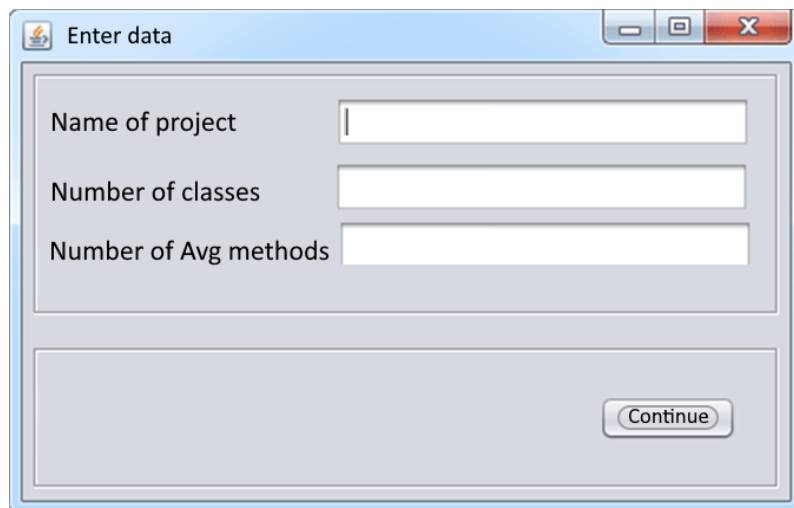


Рисунок Г.1 – Вікно для введення даних для прогнозування розміру мікросервісних e-commerce додатків на Java

З допомогою кнопки «Continue» здійснюється перехід в наступне вікно «View data» (рисунок Г.2). Це вікно призначене для перегляду введених даних для прогнозування розміру мікросервісних e-commerce додатків на Java.

Вікно «View data» містить дві кнопки «Continue» та «Return». Призначення кнопки «Continue» – перехід в наступне вікно, «Return» – повернення до початкового.

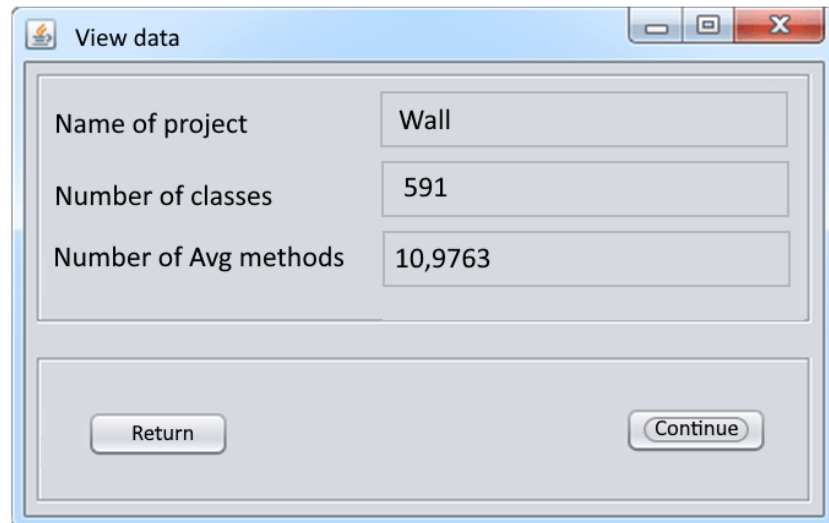


Рисунок Г.2 – Вікно для перегляду даних для прогнозування розміру мікросервісних e-commerce додатків на Java

При натисканні на кнопку «Return» повертаємося в попереднє вікно для корегування даних.

При натисканні на кнопку «Continue» переходимо в вікно «View prediction result» з вхідними даними «Name of project», «Number of classes», «Number of Avg method» та результатами прогнозування «KLOC» (оцінка розміру), «Predictional interval» (межі інтервалу прогнозування), «Confidence interval» (межі довірчого інтервалу) (Рисунок Г.3).

В вікні є дві кнопки «Return» та «Reset». При натисканні на кнопку «Return» повертаємося в перше вікно для введення даних. При натисканні на кнопку «Reset» завершуємо роботу з програмою.

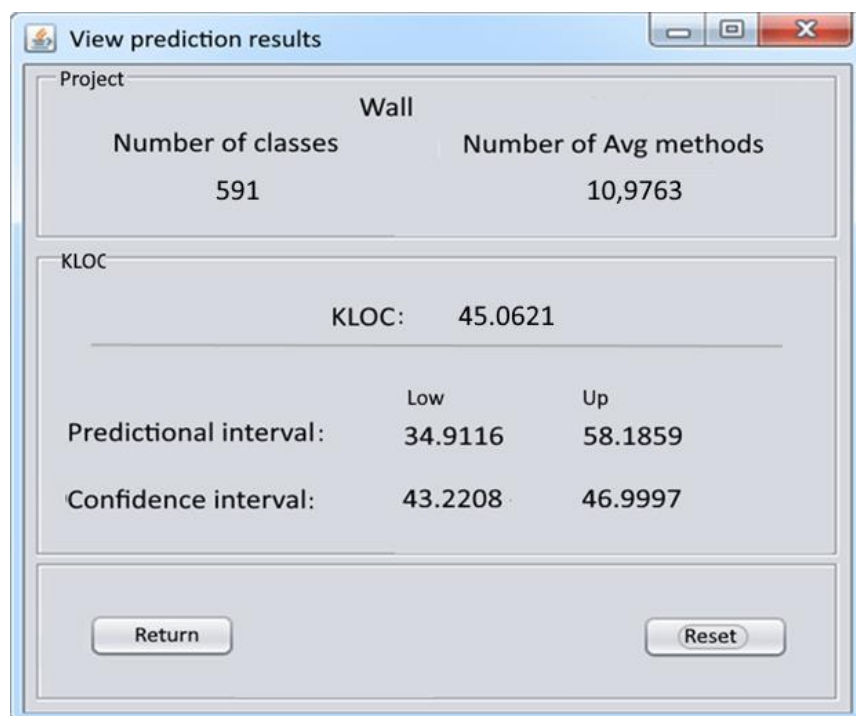


Рисунок Г.3 – Вікно для перегляду результатів прогнозування розміру мікросервісних e-commerce додатків на Java

ДОДАТОК Д

ТЕКСТ ПЗ ДЛЯ ПРОГНОЗУВАННЯ РОЗМІРУ

МІКРОСЕРВІСНИХ E-COMMERCE ДОДАТКІВ НА JAVA

JCSEException.java

```
package com.jcs.exceptions;
```

```
// клас виключень
```

```
public class JCSEException extends Exception{
```

```
    public JCSEException() {  
    }
```

```
    public JCSEException(String arg0) {  
        super(arg0);  
    }
```

```
}
```

EmptyLineException.java

```
package com.jcs.exceptions;
```

```
public class EmptyLineException extends JCSEException{
```

```
}
```

EqualsZeroException.java

```
package com.jcs.exceptions;
```

```
public class EqualsZeroException extends JCSEException{
```

```
}
```

LessThanZeroException.java

package com.jcs.exceptions;

public class LessThanZeroException extends JCSEException{

}

NoIntervalException.java

package com.jcs.exceptions;

public class NoIntervalException extends JCSEException{

}

NoSourceDataException.java

package com.jcs.exceptions;

public class NoSourceDataException extends JCSEException{

IntDP.java

package com.jcs;

import com.jcs.exceptions.*;

// клас інтервалів

public final class IntDP {

private double Imin;

private double Imax;

public double getImin() {

return Imin;

}

public double getImax() {

return Imax;

}

```

    public IntDP(double Imin, double Imax) throws LessThanZeroException,
EqualsZeroException, IllegalArgumentException{
        if(Imin <0 || Imax <0)
            throw new LessThanZeroException();
        if(Imin ==0 || Imax ==0)
            throw new EqualsZeroException();
        if(Imin >= Imax)
            throw new IllegalArgumentException();
        this. Imin = Imin;
        this. Imax = Imax;
    }
}

```

FromMathmod.java

```
package com.jcs.mat;
```

```
import com.jcs.exceptions.*;
```

```
import com.jcs.Arg;
```

```
public interface FromMathmod {
```

```

    default int    getN(){
        return 39;
    }

```

```

    default double getSzy(){
        return 0.0540;
    }

```

```

    default double getTst{
        return 2.0281;
    }

```

```
// коефіцієнти рівняння
```

```
double B0 = -1.1374;
```

```
double B1 = 0.8966;
```

```
double B2 = 0.2943;
```

```
// обернена матриця
```

```
default double[][] getMobr(){
    double Obr[][] = {
        {0.8930,0.1442},
        {0.1442,1.0768}
    };

    return Obr;
}
```

```
// середні значення
```

```
default double[] getMsred(){
    double srzn[] = {2.7703, 1.0840};

    return srzn;
}
```

```
// нормалізація даних для розрахунку
```

```
default double[] toVnormal(double x1,double x2)throws EqualsZeroException,
LessThanZeroException{
    if(x1<0.0 || x2<0.0)
        throw new EqualsZeroException();
    if(x1==0.0 || x2==0.0)
        throw new LessThanZeroException();

    double norm[] = new double[2];
```

```

    norm [0] = Math.log(x1);
    norm [1] = Math.log(x2);
    return m;
}

```

// нелінійна регресія (повертає оцінку розміру проекту)

```

    default double Predr(double x1,double x2)throws EqualsZeroException,
    LessThanZeroException{
        if(x1<0.0 || x2<0.0)
            throw new EqualsZeroException();
        if(x1==0.0 || x2==0.0)
            throw new LessThanZeroException();

        return Math.pow(10, B0)*Math.pow(x1, B1)*Math.pow(x2, B2);
    }

```

```

    default double Predr(Arg sc)throws NullPointerException{
        if(sc == null)
            throw new NullPointerException();

        return Math.pow(10, B0)*Math.pow(sc.getX1(), B1)*Math.pow(sc.getX2(), B2);
    }

```

// лінійна регресія для нормалізованих (повертає оцінку розміру проекту)

```

    default double Npredr(double x0,double x1,double x2)throws EqualsZeroException,
    LessThanZeroException{
        if(x1<0.0 || x2<0.0)
            throw new EqualsZeroException();
        if(x1==0.0 || x2==0.0)
            throw new LessThanZeroException();

        return B0+Math.log(x1)*B1+Math.log(x2)*B2;
    }

```

```

default double Npredr(Arg sc)throws NullPointerException{
    if(sc == null)
        throw new NullPointerException();

    return Math.log(B0)+Math.log(sc.getX1())*B1+Math.log(sc.getX2())*B2;
}
}

```

Arg.java

```
package com.jcs;
```

```
import com.jcs.exceptions.*;
```

```
// клас для інкапсуляції даних
```

```
public final class Arg {
```

```
    private double x1;
```

```
    public double getX1() {
        return x1;
    }

```

```
    private double x2;
```

```
    public double getX2() {
        return x2;
    }

```

```

    public Arg(double x1, double x2) throws LessThanZeroException,
EqualsZeroException{
        if(x1<0.0 || x2<0.0)
            throw new LessThanZeroException();

```

```

        if(x1==0.0 || x2==0.0)
            throw new EqualsZeroException();

        this.x1 = x1;
        this.x2 = x2;
    }

    public static Arg toArg(double x1, double x2) throws LessThanZeroException,
EqualsZeroException {

        return new Arg(x1,x2);
    }

    public double[] getNormal(){
        double norm[] = new double[2];

        norm[0] = Math.log(x1);
        norm[1] = Math.log(x2);

        return norm;
    }
}

```