

**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

«\_\_\_» \_\_\_\_\_ 2022 р.

***КВАЛІФІКАЦІЙНА РОБОТА***

**на здобуття ступеня вищої освіти «магістр»**

**на тему:** Математична модель для оцінювання складності об'єктно-орієнтованого проектування через зв'язки між класами застосунків з відкритим кодом на Java та розробка програми для її реалізації

Виконав: студент 6151м групи  
\_\_\_\_\_  
(підпис, ПІБ) Резниченко М.В.

Керівник роботи:  
Завідувач кафедри, д.т.н., професор  
(посада, науковий ступень вчене звання)  
\_\_\_\_\_  
(підпис, ПІБ) Приходько С.Б.

Миколаїв – 2022 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)

«10» жовтня 2022 р.

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

Студенту Резниченко Максиму Володимировичу

(Прізвище, ім'я, по батькові)

1. Тема роботи: Математична модель для оцінювання складності об'єктно-орієнтованого проектування через зв'язки між класами застосунків з відкритим кодом на Java та розробка програми для її реалізації

Керівник роботи проф. Приходько С.Б.

Затверджені наказом ректора № 798 уч від «28» 09 2022 р.

2. Термін подання роботи: 14.11.2022 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; Проект 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: Актуальність теми. 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. Існуючі математичні моделі для оцінювання складності ООП через зв'язки між класами. Рівняння еліпсу передбачення для нормалізованих метрик RFC та CBO. Постановка задачі на розробку проекту програми. Діаграма варіантів використання. Діаграма діяльності. Графічне вікно програми із інтерфейсними компонентами. Емпіричні дані. Результати випробувань програми. Висновки.

## 6. Консультанти розділів роботи

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

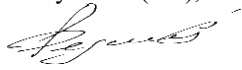
7. Дата видачі завдання 10.10.2022 р.

## КАЛЕНДАРНИЙ ПЛАН

| Номер | Назва етапів роботи                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Терміни виконання | Примітка                                                                |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|-------------------------------------------------------------------------|
| 1.    | Підготовка розділу Вступ                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 11.10.2022        | НС*                                                                     |
| 2.    | Підготовка розділу з огляду літератури та обґрунтування необхідності проведення досліджень за обраною темою                                                                                                                                                                                                                                                                                                                                                                    | 12.10.2022        | НС*                                                                     |
| 3.    | Підготовка розділу (ів) з результатів власних досліджень                                                                                                                                                                                                                                                                                                                                                                                                                       | 13.10.2022        | НС*                                                                     |
| 4.    | Підготовка розділу з проекту програмного забезпечення                                                                                                                                                                                                                                                                                                                                                                                                                          | 03.11.2022        |                                                                         |
| 5.    | Підготовка організаційно-економічного розділу                                                                                                                                                                                                                                                                                                                                                                                                                                  | 06.11.2022        |                                                                         |
| 6.    | Підготовка розділу з охорони праці                                                                                                                                                                                                                                                                                                                                                                                                                                             | 09.11.2022        |                                                                         |
| 7.    | Підготовка розділу з охорони навколишнього середовища                                                                                                                                                                                                                                                                                                                                                                                                                          | 11.11.2022        |                                                                         |
| 8.    | Підготовка розділу Висновки                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 12.11.2022        |                                                                         |
| 9.    | Оформлення списку використаних джерел та додатків                                                                                                                                                                                                                                                                                                                                                                                                                              | 13.11.2022        |                                                                         |
| 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.) | 14.11.2022        | не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку) |
| 11.   | Підготовка презентації та доповіді                                                                                                                                                                                                                                                                                                                                                                                                                                             | 19.11.2022        |                                                                         |
| 12.   | Попередній захист роботи на засіданні кафедри ПЗАС                                                                                                                                                                                                                                                                                                                                                                                                                             | 20.11.2022        |                                                                         |
| 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                                                                                                                                                                                                                                            | 28.11.2022        |                                                                         |

\* - за результатами наукового стажування (НС), яке було з 01.09.2022 по 09.10.2022 р.

Студент

  
(підпис)

Резниченко М.В.

(ПБ)

Керівник роботи

(підпис)

Приходько С.Б.

(ПБ)

## РЕФЕРАТ

Резниченко Максим Володимирович

«Математична модель для оцінювання складності об'єктно-орієнтованого проектування через зв'язки між класами застосунків з відкритим кодом на Java та розробка програми для її реалізації»

Кваліфікаційна робота на здобуття освітнього рівня магістра зі спеціальності 121 – «Інженерія програмного забезпечення». Національний університет кораблебудування імені адмірала Макарова. Миколаїв, 2022 р.

**Обсяг роботи:** 103 стор., 6 табл., 9 рис., 21 використане джерело, 5 додатків.

**Актуальність теми роботи:** необхідність підвищення достовірності оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java.

**Мета та завдання дослідження.** Метою є підвищення достовірності оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java. Завдання дослідження: проаналізувати існуючі математичні моделі для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java; удосконалити математичну модель для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java на основі нормалізуючого перетворення; розробити програму для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java.

**Об'єктом дослідження** є процес оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java.

**Предметом дослідження** є математичні моделі для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java.

**Методи дослідження.** У вирішенні поставлених завдань використано методи теорії ймовірності, математичної статистики, багатовимірного статистичного аналізу.

**Наукова новизна одержаних результатів.** Удосконалено рівняння еліпсу передбачення для десяткових логарифмів метрик RFC та CBO для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java на основі нормалізуючого перетворення у формі десяткового логарифму із врахуванням викидів у даних. Це рівняння еліпсу передбачення у порівнянні з існуючими математичними моделями враховує кореляцію між метриками RFC та CBO, що дозволяє підвищити достовірність оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java.

**Практичне значення одержаних результатів.** Розроблено програму для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java для системи моделювання Scilab 6.0.0.

**Апробація результатів досліджень.** Результати досліджень були оприлюднені на Всеукраїнській науково-практичній інтернет-конференції.

**Публікації.** Результати роботи викладені у 1 публікації, а саме: 1 матеріалах Всеукраїнської науково-практичної інтернет-конференції.

**Ключові слова:** оцінювання, складність, об'єктно-орієнтоване проектування, зв'язки між класами, застосунки з відкритим кодом, Java, еліпс передбачення.



## ABSTRACT

Maksym Volodymyrovych Reznichenko

«A mathematical model for evaluating object-oriented design complexity due to the relationships between classes of open-source applications in Java and developing the software for its implementation»

The qualification work in obtaining a master's degree in specialty 121 - "Software Engineering". Admiral Makarov National University of Shipbuilding. Mykolayiv, 2022.

**The qualification work** is presented on the 103 pages of typewritten text, contains 6 tables, 9 figures, 5 appendices and 21 references.

**Relevance of the topic of the work:** the need to increase the confidence of estimating the complexity of object-oriented design (OOD) due to the relationships between classes of open-source apps in Java.

**The goal and objectives of the study.** The goal of the study is to increase the confidence of estimating the complexity of OOD due to the relationships between classes of open-source apps in Java. The objectives of the study: to analyze existing regression models to estimate the complexity of OOD due to the relationships between classes of open-source apps in Java; to improve the mathematical model for estimating the complexity of OOD due to the relationships between classes of open-source apps in Java; to develop the software for estimating the complexity of OOD due to the relationships between classes of open-source apps in Java.

**The object of the study** is the process of estimating the complexity of OOD due to the relationships between classes of open-source apps in Java.

**The subject of the study** is the mathematical models for estimating the complexity of OOD due to the relationships between classes of open-source apps in Java.

**The methods of the study.** The methods of probability theory, mathematical statistics, multivariate statistical analysis are used in solving the problems.

**The scientific novelty of obtained results.** The prediction ellipse equation for the decimal logarithm of RFC and CBO metrics for estimating the complexity of OOD due to the relationships between classes of open-source apps in Java is improved based on the normalizing transformation in the decimal logarithm form, which allows increasing the confidence in estimating the complexity of OOD due to the relationships between classes of open-source apps in Java in comparison to existing mathematical models.

**The practical significance of obtained results** is that the software realizing the constructed model is developed for the simulation system of Scilab 6.0.0.

**Approbation of study results.** The study results were presented at the All-Ukrainian scientific Internet-conference.

**Publications.** The study results were published in one publication: the materials of the All-Ukrainian scientific Internet-conference.

**Keywords:** evaluating, complexity, object-oriented design, relationships between classes, open-source applications, Java, prediction ellipse.

## ЗМІСТ

|                                                                                                                                                                                                                       |           |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| <b>ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,<br/>СКОРОЧЕНЬ І ТЕРМІНІВ .....</b>                                                                                                                                   | <b>8</b>  |
| <b>ВСТУП .....</b>                                                                                                                                                                                                    | <b>9</b>  |
| <b>1 АНАЛІЗ ІСНУЮЧИХ МОДЕЛЕЙ ДЛЯ ОЦІНЮВАННЯ СКЛАДНОСТІ<br/>ООП ЧЕРЕЗ ЗВ'ЯЗКИ МІЖ КЛАСАМИ ЗАСТОСУНКІВ З ВІДКРИТИМ<br/>КОДОМ НА JAVA ТА ОБҐРУНТУВАННЯ НЕОБХІДНОСТІ<br/>ПРОВЕДЕННЯ ДОСЛІДЖЕНЬ ЗА ОБРАНОЮ ТЕМОЮ .....</b> | <b>13</b> |
| 1.1 Аналіз існуючих моделей для оцінювання складності ООП через<br>зв'язки між класами застосунків з відкритим кодом на Java .....                                                                                    | 13        |
| 1.2 Обґрунтування необхідності проведення досліджень за обраною<br>темою .....                                                                                                                                        | 16        |
| 1.3 Висновки за розділом 1 .....                                                                                                                                                                                      | 16        |
| <b>МАТЕМАТИЧНА МОДЕЛЬ ДЛЯ ОЦІНЮВАННЯ СКЛАДНОСТІ ООП<br/>ЧЕРЕЗ ЗВ'ЯЗКИ МІЖ КЛАСАМИ ЗАСТОСУНКІВ З ВІДКРИТИМ<br/>КОДОМ НА JAVA .....</b>                                                                                 | <b>18</b> |
| 2.1 Перевірка даних на наявність викидів для побудови математичної<br>моделі для оцінювання складності ООП через зв'язки між класами<br>застосунків з відкритим кодом на Java .....                                   | 18        |
| 2.2 Побудова рівняння еліпсу передбачення для оцінювання складності<br>ООП через зв'язки між класами застосунків з відкритим кодом на<br>Java .....                                                                   | 22        |
| 2.3 Висновки за розділом 2 .....                                                                                                                                                                                      | 24        |

|                                                                                                                                                              |           |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| <b>3 РОЗРОБКА ПРОЕКТУ ПРОГРАМИ ДЛЯ ОЦІНЮВАННЯ<br/>СКЛАДНОСТІ ООП ЧЕРЕЗ ЗВ'ЯЗКИ МІЖ КЛАСАМИ ЗАСТОСУНКІВ З<br/>ВІДКРИТИМ КОДОМ НА JAVA .....</b>               | <b>26</b> |
| 3.1 Постановка задачі на розробку проекту програми для оцінювання<br>складності ООП через зв'язки між класами застосунків з відкритим<br>кодом на Java ..... | 26        |
| 3.2 Розробка ескізного проекту програми для оцінювання складності<br>ООП через зв'язки між класами застосунків з відкритим кодом на<br>Java .....            | 28        |
| 3.3 Розробка технічного проекту програми для оцінювання складності<br>ООП через зв'язки між класами застосунків з відкритим кодом на<br>Java .....           | 34        |
| 3.4 Розробка робочого проекту програми для оцінювання складності<br>ООП через зв'язки між класами застосунків з відкритим кодом на<br>Java .....             | 38        |
| 3.5 Висновки за розділом 3 .....                                                                                                                             | 48        |
| <b>4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВПРОВАДЖЕННЯ<br/>ПРОГРАМИ .....</b>                                                                                 | <b>50</b> |
| 4.1 Розрахунок витрат на створення й експлуатацію програми .....                                                                                             | 50        |
| 4.2 Економічна ефективність розробки і впровадження програми .....                                                                                           | 53        |
| <b>5 ОХОРОНА ПРАЦІ .....</b>                                                                                                                                 | <b>56</b> |
| 5.1 Аналіз існуючої проблеми охорони праці .....                                                                                                             | 56        |
| 5.2 Аналіз шкідливих та небезпечних факторів у офісному приміщенні з<br>персональними комп'ютерами .....                                                     | 57        |
| 5.3 Розрахунок системи штучного освітлення у офісному приміщенні з<br>персональними комп'ютерами .....                                                       | 62        |

|                                                                                                                                                             |            |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <b>5.4 Розробка заходів щодо зменшення впливу шкідливих та небезпечних факторів .....</b>                                                                   | <b>65</b>  |
| <b>6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА .....</b>                                                                                                             | <b>69</b>  |
| <b>6.1 Аналіз існуючої проблеми необхідності охорони навколишнього середовища .....</b>                                                                     | <b>69</b>  |
| <b>6.2 Утилізація відходів комп'ютерної техніки .....</b>                                                                                                   | <b>72</b>  |
| <b>ВИСНОВКИ .....</b>                                                                                                                                       | <b>77</b>  |
| <b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....</b>                                                                                                                     | <b>79</b>  |
| <b>ДОДАТОК А Технічне завдання на розробку програми для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java .....</b> | <b>93</b>  |
| <b>ДОДАТОК Б Текст програми .....</b>                                                                                                                       | <b>97</b>  |
| <b>ДОДАТОК В Опис програми .....</b>                                                                                                                        | <b>106</b> |
| <b>ДОДАТОК Г Інструкція користувача .....</b>                                                                                                               | <b>107</b> |
| <b>ДОДАТОК Д Програма та методика випробувань .....</b>                                                                                                     | <b>109</b> |

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,  
СКОРОЧЕНЬ І ТЕРМІНІВ**

ООП – об’єктно-орієнтоване проектування;

ПЗ – програмне забезпечення;

CBO – Coupling Between Objects classes;

MD – the Mahalanobis distance;

OOD – object-oriented design;

RFC – the Response for Class;

SMD – the squared Mahalanobis distance.

## ВСТУП

*Актуальність теми.* На сьогодні задача оцінювання складності об'єктно-орієнтованого проектування (ООП) через зв'язки між класами є важливою для ІТ-компаній, оскільки тісно пов'язана з подальшими витратами, які виникають в процесі життєвого циклу програмного забезпечення. Згідно з [1] для оцінювання складності ООП через зв'язки між класами можуть бути використані дві програмні метрики: зв'язаність між об'єктами (CBO, Coupling Between Objects) та відгук класу (RFC, the Response for Class). Але на практиці, як правило, ці метрики аналізуються окремо без врахування кореляції між ними, а прийнятні їх значення визначаються лише на рівні класу. Так для класу, значення CBO між 1 та 5 вважається добрим [2], а рекомендоване значення RFC складає від 0 до 50 [3]. Крім того, виявлено, що високий зв'язок вказує на схильність до помилок [4]. Відповідно до [5], значення CBO більше 14 є занадто високим. Однак існують застосунки на Java, у яких значення CBO для деяких класів значно перевищують 4 і навіть 14. Наприклад, у застосунку TuxGuitar версії 1.5.2, програмні метрики якого проаналізовано в [6], є клас `org.herac.tuxguitar.app.action.installer.TGActionInstaller`, де значення CBO дорівнює 252. Зауважте, що TuxGuitar – це досить велика програма на Java з 207810 рядками коду, що включає 2381 клас. Загальний CBO для всіх класів становить 20189 [7]. Тому виникає потреба у математичній моделі, яка би дозволяла здійснювати оцінювання складності ООП через зв'язки між класами за значеннями CBO та RFC з урахуванням їх кореляції на рівні застосунку в цілому, включаючи і застосунки з відкритим кодом на Java.

*Мета і завдання дослідження.* Метою роботи є підвищення достовірності оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java.

Для досягнення поставленої мети потрібно вирішити такі завдання.

- Проаналізувати існуючі математичні моделі для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java;

- Здійснити підбор сукупності даних, за якими буде будуватися математична модель для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java, та перевірити ці дані на наявність багатовимірних викидів;

- Удосконалити математичну модель для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java на основі нормалізуючого перетворення;

- На підставі запропонованої математичної моделі розробити програму для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java.

*Об'єктом дослідження* є процес оцінювання складності об'єктно-орієнтованого проектування через зв'язки між класами застосунків з відкритим кодом на Java.

*Предметом дослідження* є математичні моделі для оцінювання складності об'єктно-орієнтованого проектування через зв'язки між класами застосунків з відкритим кодом на Java.

*Методи дослідження.* У вирішенні поставлених завдань використано методи теорії ймовірності, математичної статистики, багатовимірного статистичного аналізу.

*Наукова новизна одержаних результатів.* Удосконалено рівняння еліпсу передбачення для десяткових логарифмів метрик RFC та CBO для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java на основі нормалізуючого перетворення у формі десяткового логарифму із врахуванням викидів у даних. Зазначене рівняння у порівнянні з існуючими математичними моделями враховує як кореляцію між метриками RFC та CBO, так і те, що їх розподіл не є гаусівським. Це дозволяє підвищити достовірність оцінювання складності об'єктно-орієнтованого проектування через зв'язки між класами застосунків з відкритим кодом на Java.

*Практичне значення одержаних результатів.* Розроблено програму для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java. Програму створено scі-мовою для системи Scilab. Програму, що розроблено, можна використовувати для оцінювання складності об'єктно-орієнтованого проектування через зв'язки між класами застосунків з відкритим кодом на Java.

*Особистий внесок здобувача.* У публікації [8], що зроблена у співавторстві, автору належить рівняння еліпсу передбачення для десяткових логарифмів метрик RFC та CBO для оцінювання складності об'єктно-орієнтованого проектування через зв'язки між класами застосунків з відкритим кодом на Java, яке було побудовано під керівництвом керівника кваліфікаційної роботи проф. Приходько С.Б. У кваліфікаційній роботі також використаний матеріал доповіді «A Joint Statistical Estimation of the RFC and CBO Metrics for Open-Source Applications Developed in Java», яка була зроблена проф. Приходько С.Б. на конференції «IEEE 17th International Conference on Computer Science and Information Technologies (CSIT2022)» за секцією «Software



engineering» 11.11.2022 (<http://csit.ieee.org.ua/accepted-papers/#1631800275024-ee886edc-5296>).

*Апробація результатів досліджень.* Результати досліджень, викладені у роботі, були оприлюднені на одній конференції: III-й Всеукраїнській науково-практичній інтернет-конференції «Інформаційні технології: моделі, алгоритми, системи (ITMAS – 2022)» (Миколаїв, 26-28 жовтня 2022 р.).

*Публікації.* За результатами кваліфікаційної роботи зроблено одну публікацію, а саме: матеріали Всеукраїнської науково-практичної інтернет-конференції.

# **1 АНАЛІЗ ІСНУЮЧИХ МОДЕЛЕЙ ДЛЯ ОЦІНЮВАННЯ СКЛАДНОСТІ ООП ЧЕРЕЗ ЗВ'ЯЗКИ МІЖ КЛАСАМИ ЗАСТОСУНКІВ З ВІДКРИТИМ КОДОМ НА JAVA ТА ОБҐРУНТУВАННЯ НЕОБХІДНОСТІ ПРОВЕДЕННЯ ДОСЛІДЖЕНЬ ЗА ОБРАНОЮ ТЕМОЮ**

У першому розділі виконано аналіз існуючих моделей для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java та здійснено обґрунтування необхідності проведення досліджень за обраною темою.

## **1.1 Аналіз існуючих моделей для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java**

Як зазначалося раніше, на сьогодні задача оцінювання складності ООП через зв'язки між класами є важливою для ІТ-компаній, оскільки тісно пов'язана з подальшими витратами, які виникають в процесі життєвого циклу програмного забезпечення. Згідно з [1] для оцінювання складності ООП через зв'язки між класами можуть бути використані дві програмні метрики: зв'язаність між об'єктами (CBO, Coupling Between Objects) та відгук класу (RFC, the Response for Class). Але на практиці, як правило, ці метрики аналізуються окремо без врахування кореляції між ними, а прийнятні їх значення визначаються лише на рівні класу. Так для класу, значення CBO між 1 та 5 вважається добрим [2], а рекомендоване значення RFC складає від 0 до 50 [3]. Відповідно до [5], значення CBO більше 14 є занадто високим. Однак існують застосунки на Java, у яких значення CBO для деяких класів значно

перевищують 4 і навіть 14. Наприклад, у застосунку TuxGuitar версії 1.5.2, програмні метрики якого проаналізовано в [6], є клас `org.herac.tuxguitar.app.action.installer.TGActionInstaller`, де значення СВО дорівнює 252. Зауважте, що TuxGuitar – це досить велика програма з 207810 рядками коду, що включає 2381 клас. Загальний СВО для всіх класів становить 20189 [7]. Тому виникає потреба у математичних моделях, за допомогою яких можна було би здійснювати оцінювання складності ООП через зв'язки між класами за значеннями СВО та RFC з урахуванням їх кореляції на рівні застосунку в цілому, включаючи і застосунки з відкритим кодом на Java.

У якості такої математичної моделі, на наш погляд, можна використати еліпс передбачення. В [9] для перевірки взаємозв'язків між частинами Java-застосунків було запропоновано застосовувати рівняння трансформованого еліпсу передбачення в залежності від метрик RFC та СВО. Як відомо [10], еліпс передбачення використовують для визначення викидів у двовимірних гаусівських даних. Але двовимірний розподіл даних метрик RFC та СВО для Java-застосунків не є гаусівським. Про це свідчать наведені в [9] результати перевірки двовимірних даних метрик RFC та СВО з Java-застосунків з відкритим кодом, які розміщені на сайті GitHub (<https://github.com>), на нормальність за допомогою критерію Мардіа (Mardia's test), якій базується на багатовимірних асиметрії та ексцесі [11]. За цим критерієм двовимірний розподіл зазначених даних не є гаусівським. Тому в [9] використовувався так званий трансформований еліпс передбачення, послідовність побудови якого наведена в [12]. Трансформований еліпс передбачення будується наступним чином. Спочатку за допомогою взаємо-зворотного нормалізуючого перетворення двовимірні негаусівські дані нормалізують таким чином щоб розподіл нормалізованих даних був наближений до гаусівського. Далі для

нормалізованих даних будують еліпс передбачення. І нарешті, за еліпсом передбачення для нормалізованих даних із використанням зворотного до нормалізуючого перетворення отримують трансформований еліпс передбачення. Вихід точки негаусівських даних за трансформований еліпс передбачення буде вказувати на наявність викиду з певною довірчою ймовірністю, для якої цей еліпс було побудовано. При цьому значення довірчої ймовірності ми пропонуємо брати 0,995. Це значення зазвичай обирають при визначенні викидів у багатовимірних даних [13, 14].

Для оцінювання складності ООП через зв'язки між класами також може бути використаний еліпс передбачення для нормалізованих метрик RFC та СВО. При чому застосунок, для якого значення нормалізованих метрик RFC та СВО як координати точки будуть призводити до її виходу за межі еліпсу передбачення, буде вважатися таким, що має велику складність ООП через зв'язки між класами у порівнянні з тими застосунками, для яких було побудовано зазначений еліпс передбачення. І навпаки, застосунок буде вважатися таким, що не має великої складності ООП через зв'язки між класами у порівнянні з тими застосунками, для яких було побудовано зазначений еліпс передбачення, у протилежному випадку. При цьому у якості нормалізуючого перетворення пропонується використовувати перетворення у формі десяткового логарифму. Як і в [9] для побудови еліпсу передбачення пропонується брати значення довірчої ймовірності 0,995.

## 1.2 Обґрунтування необхідності проведення досліджень за обраною темою

Як було зазначено в попередньому підрозділі, для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java може бути застосований еліпс передбачення для нормалізованих метрик RFC та CBO. Необхідність використання саме нормалізованих метрик RFC та CBO для побудови еліпсу передбачення обумовлена тим, що двовимірний розподіл значень метрик RFC та CBO не є нормальним.

Таким чином, виникає необхідність у проведенні досліджень за обраною темою.

## 1.3 Висновки за розділом 1

У даному розділі наведено огляд існуючих математичних моделей для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java, здійснено обґрунтування необхідності проведення досліджень за обраною темою.

1) Зараз для оцінювання складності ООП через зв'язки між класами застосунків, у тому числі з відкритим кодом на Java, використовуються метрики RFC та CBO.

2) Для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java може бути застосований еліпс передбачення для нормалізованих метрик RFC та CBO. Тому виникає необхідність у подальшому удосконаленні математичної моделі для оцінювання

складності ООП через зв'язки між класами застосунків з відкритим кодом на Java.

3) Побудову еліпсу передбачення для нормалізованих метрик RFC та СВО для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java краще здійснювати із застосуванням нормалізуючих перетворень.

## **МАТЕМАТИЧНА МОДЕЛЬ ДЛЯ ОЦІНЮВАННЯ СКЛАДНОСТІ ООП ЧЕРЕЗ ЗВ'ЯЗКИ МІЖ КЛАСАМИ ЗАСТОСУНКІВ З ВІДКРИТИМ КОДОМ НА JAVA**

У другому розділі виконано перевірку даних на наявність викидів для побудови математичної моделі для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java, здійснено побудову рівняння еліпсу передбачення для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java.

### **2.1 Перевірка даних на наявність викидів для побудови математичної моделі для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java**

Для побудови математичної моделі у вигляді рівняння еліпсу передбачення для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java використовується вибірка двовимірних негаусових даних: значення метрики СВО на рівні класу  $X_1$  та значення метрики RFC на рівні класу  $X_2$  з 46 застосунків з відкритим кодом на Java. Ці дані наведені у таблиці 2.1.

Двовимірний розподіл даних з таблиці 2.1. не є гаусівським тому, що за критерієм Мардіа значення двовимірних асиметрії та ексцесу, які дорівнюють відповідно 39,18 та 15,60, більші за значення відповідних статистик 14,86 та 11,04.

Таблиця 2.1

Набір даних та квадрат відстані Махаланобіса (MD)

| №  | $X_1$  | $X_2$  | квадрат MD          |                                             |
|----|--------|--------|---------------------|---------------------------------------------|
|    |        |        | без<br>перетворення | перетворення<br>за десятковим<br>логарифмом |
| 1  | 2      | 3      | 4                   | 5                                           |
| 1  | 10,409 | 9,864  | 1,22                | 2,34                                        |
| 2  | 11,904 | 26,448 | 1,87                | 1,98                                        |
| 3  | 10,047 | 8,688  | 1,51                | 3,43                                        |
| 4  | 11,130 | 15,389 | 0,12                | 0,05                                        |
| 5  | 17,844 | 23,885 | 2,29                | 1,32                                        |
| 6  | 9,112  | 10,488 | 0,77                | 0,93                                        |
| 7  | 17,333 | 31,526 | 2,18                | 1,62                                        |
| 8  | 5,071  | 11,143 | 2,77                | 4,69                                        |
| 9  | 22,417 | 37,278 | 6,00                | 2,83                                        |
| 10 | 5,205  | 7,306  | 2,35                | 2,86                                        |
| 11 | 13,376 | 21,130 | 0,12                | 0,26                                        |
| 12 | 18,720 | 32,010 | 2,71                | 1,75                                        |
| 13 | 10,924 | 14,556 | 0,20                | 0,13                                        |
| 14 | 11,840 | 13,428 | 0,69                | 0,85                                        |
| 15 | 8,838  | 10,868 | 0,70                | 0,65                                        |
| 16 | 10,935 | 12,488 | 0,61                | 0,79                                        |
| 17 | 4,000  | 7,500  | 3,32                | 5,70                                        |
| 18 | 13,333 | 19,467 | 0,15                | 0,20                                        |
| 19 | 9,925  | 43,225 | <b>22,67</b>        | <b>13,60</b>                                |
| 20 | 10,474 | 22,947 | 1,43                | 1,80                                        |
| 21 | 8,808  | 9,077  | 1,08                | 1,80                                        |
| 22 | 7,392  | 16,351 | 1,78                | 2,59                                        |
| 23 | 14,348 | 22,215 | 0,32                | 0,41                                        |
| 24 | 9,832  | 12,969 | 0,36                | 0,22                                        |



Продовження табл. 2.1

| 1  | 2      | 3      | 4    | 5            |
|----|--------|--------|------|--------------|
| 25 | 10,245 | 15,020 | 0,16 | 0,02         |
| 26 | 18,527 | 19,751 | 4,66 | 2,59         |
| 27 | 8,625  | 14,125 | 0,58 | 0,40         |
| 28 | 15,963 | 24,214 | 0,88 | 0,74         |
| 29 | 11,912 | 14,922 | 0,35 | 0,36         |
| 30 | 11,056 | 12,653 | 0,60 | 0,77         |
| 31 | 11,970 | 18,124 | 0,00 | 0,05         |
| 32 | 10,490 | 9,521  | 1,38 | 2,81         |
| 33 | 16,028 | 25,931 | 0,94 | 0,85         |
| 34 | 11,529 | 16,265 | 0,08 | 0,04         |
| 35 | 10,077 | 18,629 | 0,44 | 0,60         |
| 36 | 17,814 | 37,158 | 4,45 | 2,67         |
| 37 | 9,103  | 12,974 | 0,44 | 0,19         |
| 38 | 10,626 | 12,596 | 0,50 | 0,58         |
| 39 | 5,437  | 5,781  | 2,36 | 3,88         |
| 40 | 16,535 | 30,537 | 1,83 | 1,49         |
| 41 | 16,590 | 16,638 | 3,55 | 2,58         |
| 42 | 2,095  | 3,264  | 5,03 | <b>13,92</b> |
| 43 | 16,883 | 31,388 | 2,09 | 1,62         |
| 44 | 13,461 | 20,525 | 0,13 | 0,23         |
| 45 | 19,838 | 28,548 | 3,47 | 1,81         |
| 46 | 8,479  | 16,069 | 0,87 | 1,01         |

Тому для виявлення викидів у даних з таблиці 2.1 ми використовуємо метод, оснований на нормалізуючих перетвореннях і квадраті відстані Махаланобіса (Mahalanobis Distance) [14, 15], яку далі позначаємо як MD. Значення квадрату MD для кожної багатовимірної точки даних  $i$ ,  $i = 1, 2, \dots, N$ , визначається

як

$$d_i^2 = (\mathbf{z}_i - \bar{\mathbf{Z}})^T S_N^{-1} (\mathbf{z}_i - \bar{\mathbf{Z}}), \quad (2.1)$$

де  $\bar{\mathbf{Z}}$  – вектор вибірових середніх;  $S_N$  – вибіркова коваріаційна матриця

$$S_N = \frac{1}{N} \sum_{i=1}^N (\mathbf{z}_i - \bar{\mathbf{Z}})(\mathbf{z}_i - \bar{\mathbf{Z}})^T. \quad (2.2)$$

Тут  $\mathbf{z}_i$  – гаусівський випадковий вектор  $\mathbf{Z} = \{Z_1, Z_2, \dots, Z_m\}^T$  для кожної багатовимірної точки даних  $i$ ,  $i = 1, 2, \dots, N$ ;  $m$  – кількість компонентів вектору  $\mathbf{Z}$ . В нашому випадку  $m$  дорівнює 2.

Вектор  $\mathbf{Z}$  отримують шляхом перетворення негаусівського випадкового вектору  $\mathbf{X} = \{X_1, X_2, \dots, X_m\}^T$  за допомогою багатовимірного нормалізуючого перетворення

$$\mathbf{Z} = \psi(\mathbf{X}). \quad (2.3)$$

В нашому випадку у якості перетворення (2.3) ми використовуємо перетворення у вигляді десяткового логарифму.

Відомо [13], якщо дані мають багатовимірний нормальний розподіл, то розподіл квадрату MD поводитья як розподіл  $\chi^2$ . Для великих  $N - m$  (принаймні більше 25), відстань  $d_i^2$  повинна вести себе приблизно як незалежні випадкові величини  $\chi_{m,\alpha}^2$ . Тут  $\chi_{m,\alpha}^2$  – це квантиль розподілу  $\chi^2$  з рівнем значимості  $\alpha$ . Значення даних, для яких величина квадрату MD більша, ніж

квантиль розподілу  $\chi^2$ , розглядаються як викиди, і ці значення відкидаються. Після відкидання зменшене число багатомірних точок даних знову нормалізується, використовуючи перетворення (2.3), доки всі значення квадрату MD (2.1) будуть меншими чи дорівнювати квантилю розподілу  $\chi^2$ .

Для рівня значимості 0,005 та перетворення у вигляді десяткового логарифму є два викиди (рядки 19 та 42) тому, що для цих рядків даних квадрат MD є більшим ніж величина квантіля розподілу  $\chi^2$ , яка становить 10,60 для рівня значимості 0,005. Зауважимо, застосування критерію на основі квадрату MD до даних з таблиці 2.1 без їх нормалізації показало, що лише дані рядка 19 є двовимірним викидом, оскільки для цього рядка даних квадрат MD дорівнює 22,67 і є більшим ніж величина квантіля розподілу  $\chi^2$ , яка становить 10,60 для рівня значимості 0,005.

В подальшому двовимірні негаусові дані, що наведені у таблиці 2.1, без рядків 19 та 42 використовуються для побудови рівняння еліпса передбачення для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java.

## 2.2 Побудова рівняння еліпсу передбачення для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java

Для побудови рівняння еліпсу передбачення для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java спочатку двовимірні негаусівські дані з таблиці 2.1 ми нормалізуємо за допомогою

перетворення у вигляді десяткового логарифму. Далі рівняння для еліпса передбачення для нормалізованих метрик RFC і СВО визначається як [16]

$$(\mathbf{Z} - \bar{\mathbf{Z}})^T \mathbf{S}_N^{-1} (\mathbf{Z} - \bar{\mathbf{Z}}) = \frac{2(N^2 - 1)}{N(N - 2)} F_{2, N-2, \alpha}, \quad (2.4)$$

де  $\mathbf{Z}$  – це гаусівський випадковий вектор,  $\mathbf{Z} = \{Z_1, Z_2\}^T$ ;  $\bar{\mathbf{Z}}$  – вектор вибірових середніх,  $\bar{\mathbf{Z}} = \{\bar{Z}_1, \bar{Z}_2\}^T$ ;  $N$  – кількість точок даних (у нашому випадку  $N=44$ );  $F_{2, N-2, \alpha}$  – квантиль  $F$  розподілу з 2 та  $N-2$  ступенями свободи;  $\alpha$  – рівень значущості;  $\mathbf{S}_N$  – вибіркова коваріаційна матриця

$$\mathbf{S}_N = \frac{1}{N} \sum_{i=1}^N (\mathbf{Z}_i - \bar{\mathbf{Z}})(\mathbf{Z}_i - \bar{\mathbf{Z}})^T, \quad (2.5)$$

У нашому випадку матриця (2.5) є такою:

$$\mathbf{S}_N = \begin{pmatrix} 0,02666 & 0,02689 \\ 0,02689 & 0,03889 \end{pmatrix}. \quad (2.6)$$

Для матриці (2.6) обернена матриця для рівняння (2.4) є наступною:

$$\mathbf{S}_N^{-1} = \begin{pmatrix} 123,84 & -85,62 \\ -85,62 & 84,91 \end{pmatrix}. \quad (2.7)$$

У нашому випадку вектор вибірових середніх  $\bar{\mathbf{Z}}$  визначається як  $\bar{\mathbf{Z}} = \{1,056; 1,215\}^T$ . Ми приймаємо рівень значущості  $\alpha$  як 0,005 згідно з [17].

Зауважте, що ліва частина рівняння (2.4) є SMD, яка для кожної точки двовимірних даних  $i$ , визначається за (2.1). Відомо [18], що тестова статистика для SMD  $d_i^2$  може бути створена наступним чином

$$N(N-2)d_i^2/2(N^2-1), \quad (2.8)$$

яка приблизно має  $F$  розподіл з 2 та  $N-2$  ступенями свободи.

Ми використали тестову статистику (2.8) для побудови рівняння (2.4) еліпса передбачення для нормалізованих метрик RFC та CBO.

Ми можемо застосувати рівняння (2.4), щоб допомогти дизайнерам ПЗ в оцінці складності на третьому етапі ООП застосунків, що розробляються на Java.

### 2.3 Висновки за розділом 2

У даному розділі виконано перевірку даних на наявність викидів для побудови математичної моделі для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java, здійснено побудову рівняння еліпсу передбачення для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java.

1) Перевірку даних для побудови рівняння еліпсу передбачення для оцінювання складності ООП через зв'язки між класами застосунків з відкритим

кодом на Java на наявність викидів виконано за методом, який оснований на нормалізуючих перетвореннях і квадраті відстані Махаланобіса (Mahalanobis Distance). З'ясовано, що за методом, який оснований на нормалізуючих перетвореннях і квадраті відстані Махаланобіса, двовимірні дані містять два двовимірних викиди. Застосування критерія на основі квадрата відстані Махаланобіса до даних без їх нормалізації показало, що дані містять лише один двовимірний викид. Такий результат пояснюється тим, що зазначені дані мають розподіл, який не є гаусівським.

2) Удосконалено рівняння еліпсу передбачення для десяткових логарифмів метрик RFC та CBO для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java на основі нормалізуючого перетворення у формі десяткового логарифму із врахуванням викидів у даних. Зазначене рівняння у порівнянні з існуючими математичними моделями враховує як кореляцію між метриками RFC та CBO, так і те, що їх розподіл не є гаусівським. Це дозволяє підвищити достовірність оцінювання складності об'єктно-орієнтованого проектування через зв'язки між класами застосунків з відкритим кодом на Java.

3) Отримані результати свідчать про те, що для побудови рівняння еліпсу передбачення для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java, потрібно використовувати нормалізуючі перетворення з врахуванням викидів у даних.

### **3 РОЗРОБКА ПРОЕКТУ ПРОГРАМИ ДЛЯ ОЦІНЮВАННЯ СКЛАДНОСТІ ООП ЧЕРЕЗ ЗВ'ЯЗКИ МІЖ КЛАСАМИ ЗАСТОСУНКІВ З ВІДКРИТИМ КОДОМ НА JAVA**

У третьому розділі розроблено проект програми для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java. Здійснено постановку задачі на розробку проекту програми, виконано ескізне, технічне та робоче проєктування.

#### **3.1 Постановка задачі на розробку проекту програми для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java**

Виходячи з аналізу існуючих моделей для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java, сформулюємо постановку задачі. Потрібно створити програму для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java, яка би задовольняла таким вимогам.

#### **Вимоги до програми.**

*Вимоги функціональних характеристик.*

Вимоги до складу функцій, що виконуються.

Програма має виконувати такі функції:

1) Вводити з текстового файлу емпіричні дані з метрик СВО та RFC застосунків з відкритим кодом на Java.

2) Вводити з інтерфейсу програми довірчу ймовірність, метрики CBO та RFC, за якими потрібно оцінити складність ООП застосунку через зв'язки між класами із заданою ймовірністю.

3) Виконувати нормалізацію емпіричних даних за нормалізуючим перетворенням у формі десяткового логарифму.

4) Здійснювати побудову рівняння еліпсу передбачення для нормалізованих метрик CBO та RFC.

5) Обчислювати значення тестової статистики для квадрата відстані Махаланобіса.

7) Здійснювати оцінювання складності ООП застосунку через зв'язки між класами із заданою ймовірністю.

8) Виводити на екран значення квадрата відстані Махаланобіса для нормалізованих метрик CBO та RFC, значення тестової статистики для квадрата відстані Махаланобіса, результати оцінювання складності ООП застосунку через зв'язки між класами із заданою ймовірністю.

9) Виводити у текстовий файл значення квадрата відстані Махаланобіса для нормалізованих метрик CBO та RFC, значення тестової статистики для квадрата відстані Махаланобіса, результати оцінювання складності ООП застосунку через зв'язки між класами із заданою ймовірністю.

*Вимоги до складу й параметрів технічних засобів.*

Програма має експлуатуватись на обладнанні з параметрами не нижчими за наступні: процесор Pentium IV (або еквівалентний) з інструкціями SSE2, оперативна пам'ять 2 Гб (1 Гб мінімум), простір на жорсткому диску 600 Мб мінімум.



*Вимоги до інформаційної та програмної сумісності.*

Для запуску та роботи з програмою необхідна операційна система Windows Vista/7/8/10 (32 або 64 біти), система моделювання Scilab 6.0.0., компілятор C (Visual Studio 2010 або Visual Express 2010) для компіляції зовнішніх модулів C або C++ та використання моделі в Xcos.

На підставі цієї постановки задачі розроблено технічне завдання, яке наведено у додатку А.

### 3.2 Розробка ескізного проекту програми для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java

В ескізному проекті було виявлено варіанти використання та побудовано їх сценарії, розроблено зовнішню специфікацію програми для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java та стани цієї програми при виконанні.

Даний проект передбачає наявність лише одного актора – користувача програми, який буде повноцінно використовувати всі функції програми. Виявлені варіанти використання знаходяться в таблиці 3.1.

Діаграма варіантів використання програми для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java зображена на рисунку 3.1.

Таблиця 3.1 Виявлені варіанти використання

| Основні актори | Найменування                            | Формулювання                                                                                                                                                           |
|----------------|-----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Користувач     | Ввести дані метрик СВО та RFC           | Вводить дані метрик СВО та RFC із застосунків з відкритим кодом на Java.                                                                                               |
| Користувач     | Ввести метрики СВО та RFC застосунку    | Вводить метрики СВО та RFC, за якими потрібно оцінити складність ООП застосунку через зв'язки між класами.                                                             |
| Користувач     | Ввести довірчу ймовірність              | Вводить довірчу ймовірність, для якої потрібно оцінити складність ООП застосунку через зв'язки між класами.                                                            |
| Користувач     | Нормалізувати дані                      | Виконує нормалізацію внесених даних.                                                                                                                                   |
| Користувач     | Побудувати рівняння еліпсу передбачення | Будує рівняння еліпсу передбачення для нормалізованих метрик СВО та RFC для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java. |
| Користувач     | Оцінити складність ООП застосунку       | Оцінює складність ООП застосунку через зв'язки між класами із заданою ймовірністю.                                                                                     |

Сценарії варіантів використання зображені у вигляді діаграми діяльності на рисунку 3.2.

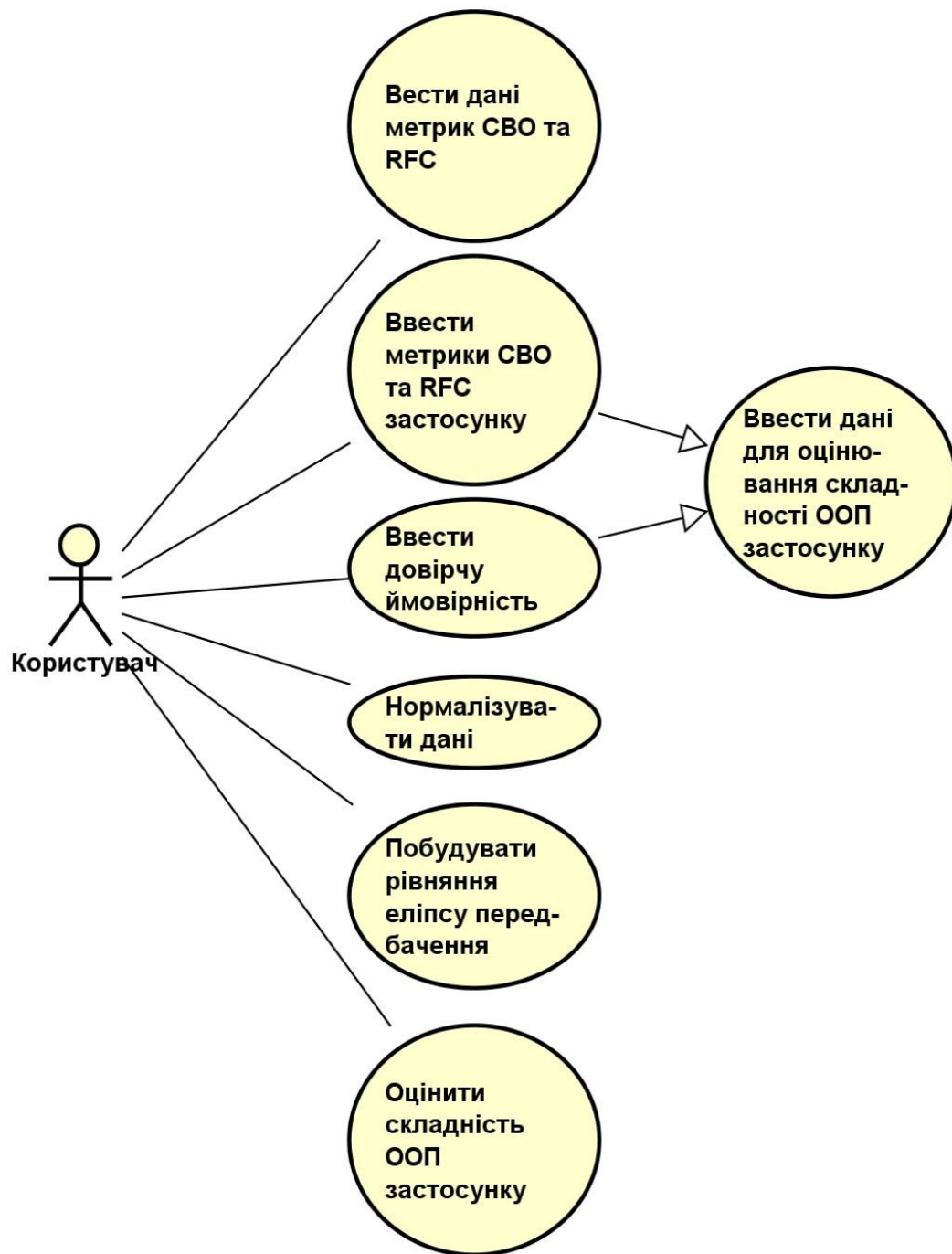


Рисунок 3.1 – Діаграма варіантів використання програми для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java



Рисунок 3.2 – Діаграма діяльності ПЗ для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java

Зовнішня специфікація програми.

*Назва програми:* Pred\_El\_lg\_CBO\_RFC.

*Функції, які виконує програма:*

1) Вводити з текстового файлу емпіричні дані з метрик CBO та RFC застосунків з відкритим кодом на Java.

2) Вводити з інтерфейсу програми довірчу ймовірність, метрики CBO та RFC, за якими потрібно оцінити складність ООП застосунку через зв'язки між класами із заданою ймовірністю.

3) Виконувати нормалізацію емпіричних даних за нормалізуючим перетворенням у формі десяткового логарифму.

4) Здійснювати побудову рівняння еліпсу передбачення для нормалізованих метрик CBO та RFC.

5) Обчислювати значення тестової статистики для квадрата відстані Махаланобіса.

7) Здійснювати оцінювання складності ООП застосунку через зв'язки між класами із заданою ймовірністю.

8) Виводити на екран значення квадрата відстані Махаланобіса для нормалізованих метрик CBO та RFC, значення тестової статистики для квадрата відстані Махаланобіса, результати оцінювання складності ООП застосунку через зв'язки між класами із заданою ймовірністю.

9) Виводити у текстовий файл значення квадрата відстані Махаланобіса для нормалізованих метрик CBO та RFC, значення тестової статистики для квадрата відстані Махаланобіса, результати оцінювання складності ООП застосунку через зв'язки між класами із заданою ймовірністю.

*Вхідні дані:*

- емпіричні дані з метрик CBO та RFC застосунків з відкритим кодом на Java;

- значення довірчої ймовірності, значення метрик CBO та RFC, за якими потрібно оцінити складність ООП застосунку через зв'язки між класами із заданою ймовірністю

*Вихідні дані:*

- значення квадрата відстані Махаланобіса для нормалізованих метрик CBO та RFC;

- значення тестової статистики для квадрата відстані Махаланобіса.

*Зовнішні ефекти:* поява графічного вікна для вводу даних, управління виводом даних, графічного вікна для виводу даних, вихід з програми (закриття всіх графічних вікон).

Для проектування інтерфейсу програми для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java представимо стани програми при виконанні у вигляді відповідної діаграми станів. Діаграма станів програми для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java наведена на рисунку 3.3.

На діаграмі станів програми введені такі позначення станів: 0 – початковий стан, 1 – графічне вікно головного модуля, 2 – графічне вікно візуалізації введених емпіричних даних, 3 – графічне вікно із результатами оцінювання складності ООП через зв'язки між класами застосунку з відкритим кодом на Java, 4 – графічне вікно з результатами побудови еліпсу передбачення для нормалізованих метрик CBO та RFC для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java, 5 – кінцевий стан.

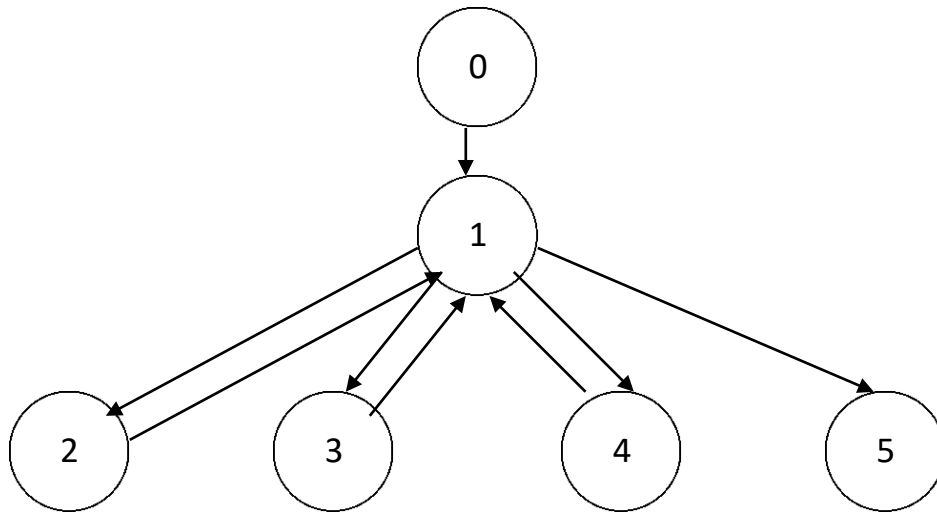


Рисунок 3.3 – Діаграма станів програми для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java

Далі було здійснено розробку технічного проекту програми для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java.

### 3.3 Розробка технічного проекту програми для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java

В технічному проекті було спроектовано програму для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java на основі покрокової деталізації зверху вниз, розроблено специфікації модулів програми, побудовано алгоритми модулів програми.

Програма складається з таких модулів: головного модуля (Pred\_El\_lg\_CBO\_RFC), модуля для введення емпіричних даних з файлу та їх візуалізації (DataInput), модуля для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java (JavaAppComplEval), модуля для закриття графічних вікон (W\_Close).

Специфікація головного модуля.

*Назва модуля:* Pred\_El\_lg\_CBO\_RFC.

*Функції, які виконує програма:*

- 1) Відкриття графічного вікна головного модуля.
- 2) Введення імені файлу з емпіричними даними.
- 3) Виклик функції для введення емпіричних даних з файлу та їх візуалізації у графічному вікні.
- 4) Вибір наявності нормалізації.
- 5) Виклик функції для оцінювання складності ООП через зв'язки між класами застосунку з відкритим кодом на Java.
- 6) Виклик функції для закриття графічних вікон.

*Вхідні дані:*

- емпіричні дані з метрик CBO та RFC застосунків з відкритим кодом на Java;
- значення довірчої ймовірності, значення метрик CBO та RFC, за якими потрібно оцінити складність ООП застосунку через зв'язки між класами із заданою ймовірністю.

*Вихідні дані:*

- значення квадрата відстані Махаланобіса для нормалізованих метрик CBO та RFC;
- значення тестової статистики для квадрата відстані Махаланобіса.



*Зовнішні ефекти:* поява графічного вікна для вводу даних, управління виводом даних, графічного вікна для виводу даних, вихід з програми (закриття всіх графічних вікон).

Специфікація модуля для введення емпіричних даних з файлу та їх візуалізації.

*Назва модуля:* DataInput.

*Функції, які виконує модуль:*

- 1) Введення емпіричних даних з файлу.
- 2) Формування одномірних та двомірних масивів значень випадкових величини (емпіричних даних).
- 3) Відкриття графічного вікна для візуалізації введених емпіричних даних.
- 4) Візуалізацію введених емпіричних даних.

*Вхідні дані:*

- значення випадкових величини (емпіричні дані).

*Вихідні дані:*

- одномірні та двомірні масиви значення випадкових величини (емпіричні дані);
- кількість значень випадкових величин (емпіричних даних);
- ідентифікатор графічного вікна.

*Зовнішні ефекти:* Поява графічного вікна для візуалізації введених емпіричних даних.

Специфікація модуля для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java

*Назва модуля:* JavaAppComplEval.

*Функції, які виконує модуль:*

1) Вводить з інтерфейсу програми довірчу ймовірність, метрики CBO та RFC, за якими потрібно оцінити складність ООП застосунку через зв'язки між класами із заданою ймовірністю.

2) Виконує нормалізацію емпіричних даних за нормалізуючим перетворенням у формі десяткового логарифму.

3) Здійснює побудову рівняння еліпсу передбачення для нормалізованих метрик CBO та RFC.

4) Обчислює значення тестової статистики для квадрата відстані Махаланобіса.

5) Здійснює оцінювання складності ООП застосунку через зв'язки між класами із заданою ймовірністю.

6) Виводить на екран значення квадрата відстані Махаланобіса для нормалізованих метрик CBO та RFC, значення тестової статистики для квадрата відстані Махаланобіса, результати оцінювання складності ООП застосунку через зв'язки між класами із заданою ймовірністю.

9) Виводить у текстовий файл значення квадрата відстані Махаланобіса для нормалізованих метрик CBO та RFC, значення тестової статистики для квадрата відстані Махаланобіса, результати оцінювання складності ООП застосунку через зв'язки між класами із заданою ймовірністю.

*Вхідні дані:*

- емпіричні дані з метрик CBO та RFC застосунків з відкритим кодом на Java;

- значення довірчої ймовірності, значення метрик CBO та RFC, за якими потрібно оцінити складність ООП застосунку через зв'язки між класами із заданою ймовірністю.

*Вихідні дані:*

- значення квадрата відстані Махаланобіса для нормалізованих метрик CBO та RFC;

- значення тестової статистики для квадрата відстані Махаланобіса.

*Зовнішні ефекти:* з'являються значення квадрата відстані Махаланобіса для нормалізованих метрик CBO та RFC, значення тестової статистики для квадрата відстані Махаланобіса, результати оцінювання складності ООП застосунку через зв'язки між класами із заданою ймовірністю.

*Назва модуля:* W\_Close.

*Функції, які виконує модуль:*

Закриття графічних вікон.

*Вхідні дані:*

- ідентифікатори графічних вікон.

*Вихідні дані:* відсутні.

*Зовнішні ефекти:* Закриття всіх графічних вікон.

В подальшому були розроблені алгоритми зазначених модулів.

### 3.4 Розробка робочого проекту програми для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java

В робочому проекті було розроблено програму для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java, здійснено тестування програми, розроблено програмну документацію, здійснено випробування програми.

Спочатку sci-мовою системи моделювання Scilab було виконано кодування алгоритмів, що були розроблені на стадії технічного проекту. Текст програми наведений в додатку Б. Приклад кодування модуля JavaAppComplEval, що здійснює оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java, sci-мовою в системі моделювання Scilab наведено нижче.

```
//Функція оцінювання складності ООП через зв'язки між класами застосунків
з відкритим кодом на Java
function JavaAppComplEval(x, n)
//Считываем значение переменных из текстовых полей и
//преобразовываем их к числовому типу.
x1i=eval(get(edit_text4,'string'));
x2i=eval(get(edit_text5,'string'));
//x3i=eval(get(edit_text6,'string'));
Pconf=eval(get(edit_text6,'string'));

x1A=0; x2A=0;
//x3A=0; yA=0;
for i=1:n
    x1A=x1A+x(i,1); x2A=x2A+x(i,2);
    //    x3A=x3A+x(i,3); yA=yA+yx(i,1);
end;
x1A=x1A/n; x2A=x2A/n;
//x3A=x3A/n; yA=yA/n;
Sx1x1=0; Sx2x2=0;
//Sx3x3=0;
Sx1x2=0;
//Sx1x3=0; Sx2x3=0; SyA=0;
for i=1:n
    Sx1x1=Sx1x1+(x(i,1)-x1A)*(x(i,1)-x1A);
    Sx1x2=Sx1x2+(x(i,1)-x1A)*(x(i,2)-x2A);
    //    Sx1x3=Sx1x3+(x(i,1)-x1A)*(x(i,3)-x3A);
    //    Sx2x3=Sx2x3+(x(i,2)-x2A)*(x(i,3)-x3A);
    Sx2x2=Sx2x2+(x(i,2)-x2A)*(x(i,2)-x2A);
    //    Sx3x3=Sx3x3+(x(i,3)-x3A)*(x(i,3)-x3A);
    //    SyA=SyA+(yx(i,1)-yA)*(yx(i,1)-yA);
end;
Sxx(1,1)=Sx1x1; Sxx(1,2)=Sx1x2;
//Sxx(1,3)=Sx1x3;
Sxx(2,1)=Sx1x2; Sxx(2,2)=Sx2x2;
//Sxx(2,3)=Sx2x3;
//Sxx(3,1)=Sx1x3; Sxx(3,2)=Sx2x3; Sxx(3,3)=Sx3x3;

Sxx_1=1/Sxx;
```

```

z1A=0; z2A=0;
//z3A=0;
for i=1:n
    z1A=z1A+log10(x(i,1)); z2A=z2A+log10(x(i,2));
    //    z3A=z3A+log10(x(i,3));
end;
z1A=z1A/n; z2A=z2A/n;
//z3A=z3A/n;
Sz1z1=0; Sz2z2=0;
//Sz3z3=0;
Sz1z2=0;
//Sz1z3=0; Sz2z3=0;
for i=1:n
    Sz1z1=Sz1z1+(log10(x(i,1))-z1A)*(log10(x(i,1))-z1A);
    Sz1z2=Sz1z2+(log10(x(i,1))-z1A)*(log10(x(i,2))-z2A);
    //    Sz1z3=Sz1z3+(log10(x(i,1))-z1A)*(log10(x(i,3))-z3A);
    //    Sz2z3=Sz2z3+(log10(x(i,2))-z2A)*(log10(x(i,3))-z3A);
    Sz2z2=Sz2z2+(log10(x(i,2))-z2A)*(log10(x(i,2))-z2A);
    //    Sz3z3=Sz3z3+(log10(x(i,3))-z3A)*(log10(x(i,3))-z3A);
end;
Szz(1,1)=Sz1z1/n; Szz(1,2)=Sz1z2/n;
//Szz(1,3)=Sz1z3;
Szz(2,1)=Sz1z2/n; Szz(2,2)=Sz2z2/n;
//Szz(2,3)=Sz2z3;
//Szz(3,1)=Sz1z3; Szz(3,2)=Sz2z3; Szz(3,3)=Sz3z3;

Szz_1=1/Szz;

zi(1)=log10(x1i)-z1A; zi(2)=log10(x2i)-z2A;
//zi(3)=log10(x3i)-z3A;
ziT=zi';
ziSzz_1=ziT*Szz_1;
result=ziSzz_1*zi;

Dfn=2;
Dfd=n-2;
//    Q=0.005;
//    P=1-Q;
Q=1-Pconf;
P=Pconf;
[F]=cdf('F',Dfn,Dfd,P,Q);
test_st=2*(n*n-1)*F/n/(n-2);

set(textresult,'string', sprintf("=%1.4f",result));
set(textresult2,'string', sprintf("=%1.4f",test_st));
//set(textresult3,'string', sprintf("=%1.4f",yUBC));
//set(textresult4,'string',...
//    sprintf("Коефіцієнт детермінації R^2=%1.4f",r_2));

```

```

//set(textresult5,'string', sprintf("=%1.4f",yLBPr));
//set(textresult6,'string', sprintf("=%1.4f",yUBPr));

if result > test_st then
    set(textresult4,'string',sprintf("Складність висока
SMD=%1.4f",result));
else
    set(textresult4,'string',sprintf("Складність не висока
SMD=%1.4f",result));
end;

if (get(audio_bikv_x,'value')==0)

else

end;

endfunction

```

Далі в середовищі Scilab 6.0.0 було створено sse-файл розміром 10 КБ. Зазначимо, що в програмі використовувався динамічний спосіб створення інтерфейсних компонентів. Він полягає у тому, що при виконанні програми можуть створюватися і знищуватися ті чи інші елементи управління (командні кнопки, радіокнопки, мітки та таке інше), а їх властивостям присвоюються відповідні значення. В програмі були використані такі інтерфейсні компоненти: вікно редагування, командні кнопки та радіокнопки. Графічне вікно програми із зазначеними інтерфейсними компонентами наведено на рис.3.4.

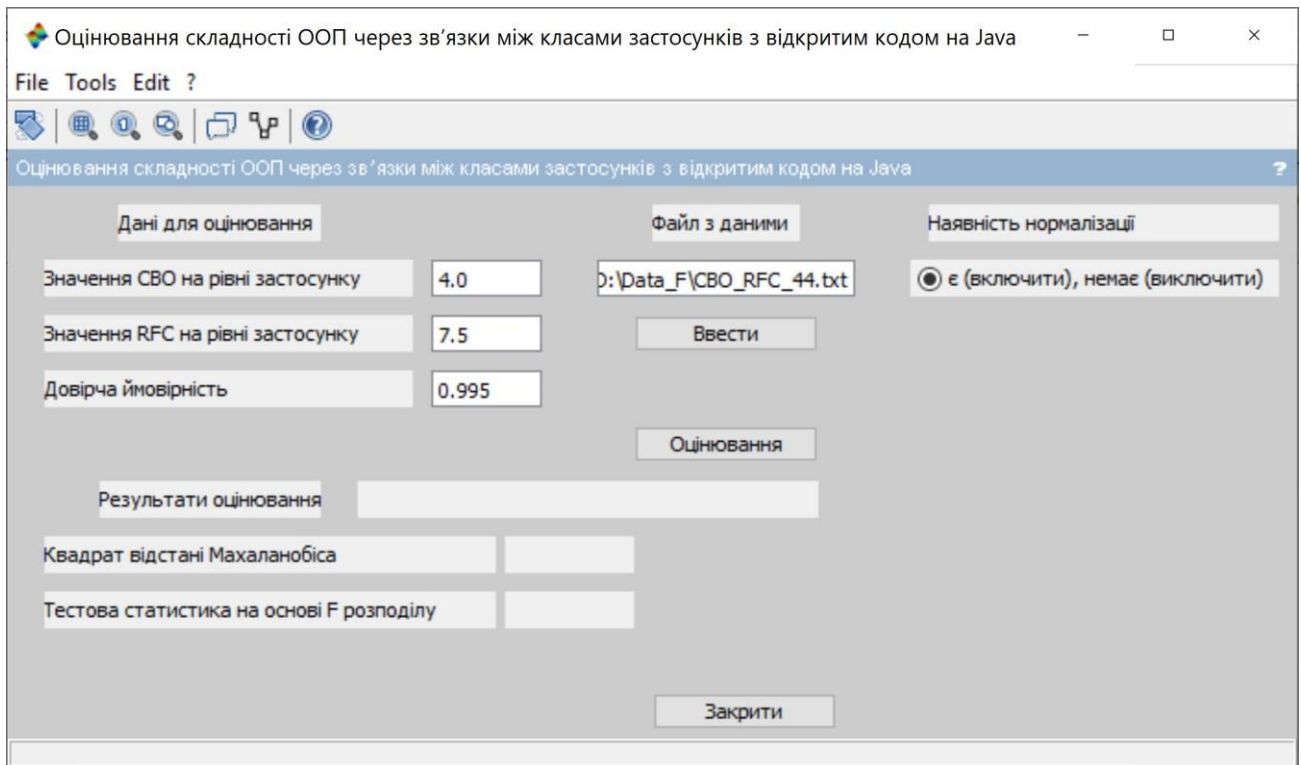


Рисунок 3.4. – Графічне вікно програми із інтерфейсними компонентами

Було проведено тестування окремих модулів. У якості прикладу наведемо результати тестування для модуля оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java для емпіричних даних з метрик СВО та RFC 44 застосунків з відкритим кодом на Java. Було побудовано рівняння еліпсу передбачення на основі нормалізуючого перетворення у формі десятикового логарифму. Результати візуалізації емпіричних даних з метрик СВО та RFC 44 застосунків з відкритим кодом на Java наведені на рис.3.5.

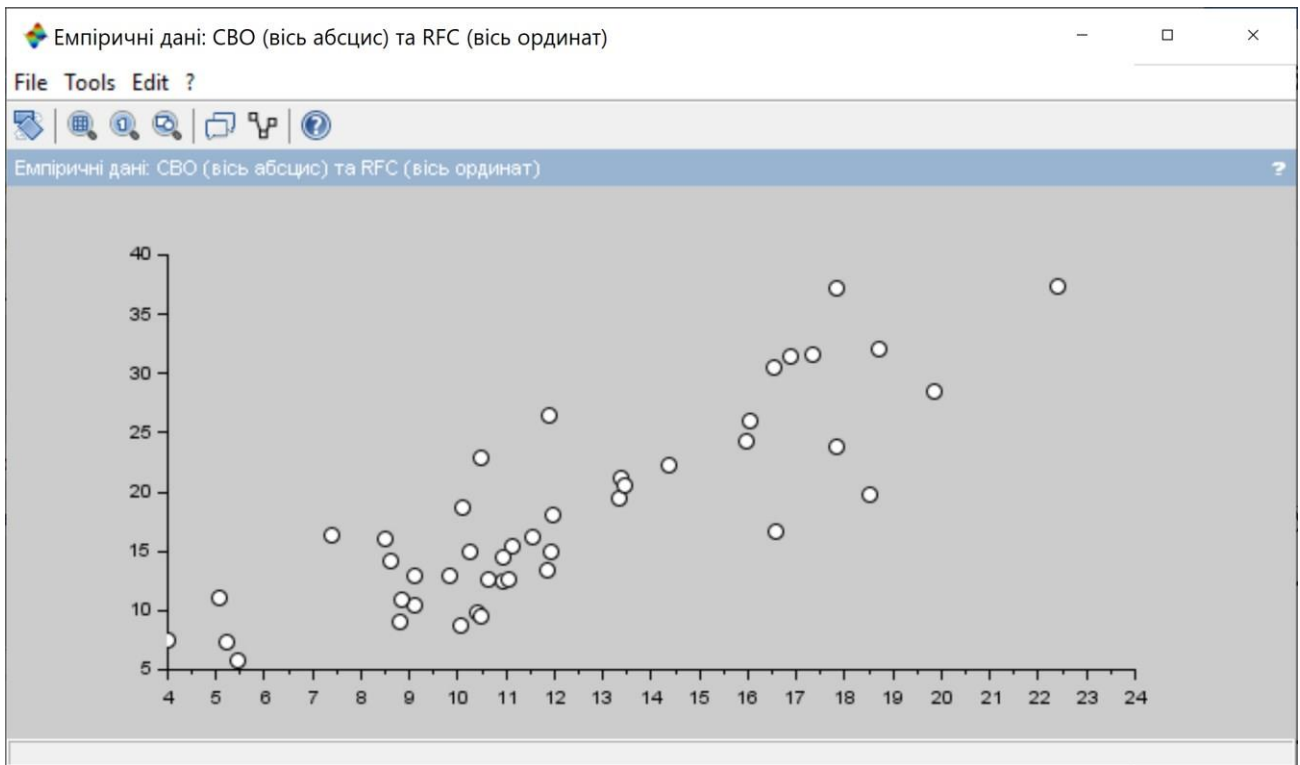


Рисунок 3.5. – Емпіричні дані з метрик CBO та RFC 44 застосунків з відкритим кодом на Java

Результати оцінювання складності ООП через зв'язки між класами застосунку TuxGuitar з відкритим кодом на Java, результати обчислення квадрата відстані Махаланобіса для нормалізованих метрик CBO та RFC для цього застосунку, значення тестової статистики для квадрата відстані Махаланобіса наведені на рис.3.6. Зауважте, що TuxGuitar, програмні метрики якого проаналізовано в [6], є досить великою програмою з 207810 рядками коду, що включає 2381 клас для версії 1.5.2.



Оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java

File Tools Edit ?

Оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java

Дані для оцінювання

Значення СВО на рівні застосунку: 7.32

Значення RFC на рівні застосунку: 17.78

Довірча ймовірність: 0.995

Файл з даними: D:\Data\_F\CBO\_RFC\_44.txt

Ввести

Наявність нормалізації: ☒ є (включити), ☐ немає (виключити)

Оцінювання

Результати оцінювання

Складність не висока SMD=5.7865

Квадрат відстані Махаланобіса: =5.7865

Тестова статистика на основі F розподілу: =12.6207

Закрити

Рисунок 3.6. – Результати оцінювання складності ООП через зв'язки між класами застосунку TuxGuitar

Результати оцінювання складності ООП через зв'язки між класами застосунку Metro\_systems [19] з відкритим кодом на Java, результати обчислення квадрата відстані Махаланобіса для нормалізованих метрик СВО та RFC для цього застосунку, значення тестової статистики для квадрата відстані Махаланобіса наведені на рис.3.7.

Оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java

File Tools Edit ?

Оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java

| Дані для оцінювання              |       | Файл з даними            | Наявність нормалізації                                           |
|----------------------------------|-------|--------------------------|------------------------------------------------------------------|
| Значення СВО на рівні застосунку | 9.93  | D:\Data_F\CBO_RFC_44.txt | <input checked="" type="radio"/> є (включити), немає (виключити) |
| Значення RFC на рівні застосунку | 43.23 |                          |                                                                  |
| Довірча ймовірність              | 0.995 |                          |                                                                  |

Ввести

Оцінювання

| Результати оцінювання                    |          |
|------------------------------------------|----------|
| Складність висока SMD                    | =19.7108 |
| Квадрат відстані Махаланобіса            | =19.7108 |
| Тестова статистика на основі F розподілу | =12.6207 |

Закрити

Рисунок 3.7. – Результати оцінювання складності ООП через зв'язки між класами застосунку Metro\_systems

Metro\_systems – це Core Java проект з управління системами метро та впровадження смарт-карт для щоденних користувачів [19]. Як ми бачимо (рис. 3.7), Metro\_systems має велику складність ООП через зв'язки між класами. В цьому можна пересвідчитися і з рис. 3.8, на якому зображено у вигляді хреста вихід точки з координатами, які відповідають значенням нормалізованих метрик RFC та СВО застосунку Metro\_systems, за еліпс передбачення (червона лінія).

А як зазначалося раніше, застосунок, для якого значення нормалізованих метрик RFC та СВО як координати точки призводять до її виходу за межі еліпсу

передбачення, вважається таким, що має велику складність ООП через зв'язки між класами у порівнянні з тими застосунками, для яких було побудовано зазначений еліпс передбачення. І навпаки, застосунок буде вважатися таким, що не має великої складності ООП через зв'язки між класами у порівнянні з тими застосунками, для яких було побудовано зазначений еліпс передбачення, у протилежному випадку. Саме цей останній випадок ми спостерігаємо на рис. 3.9.

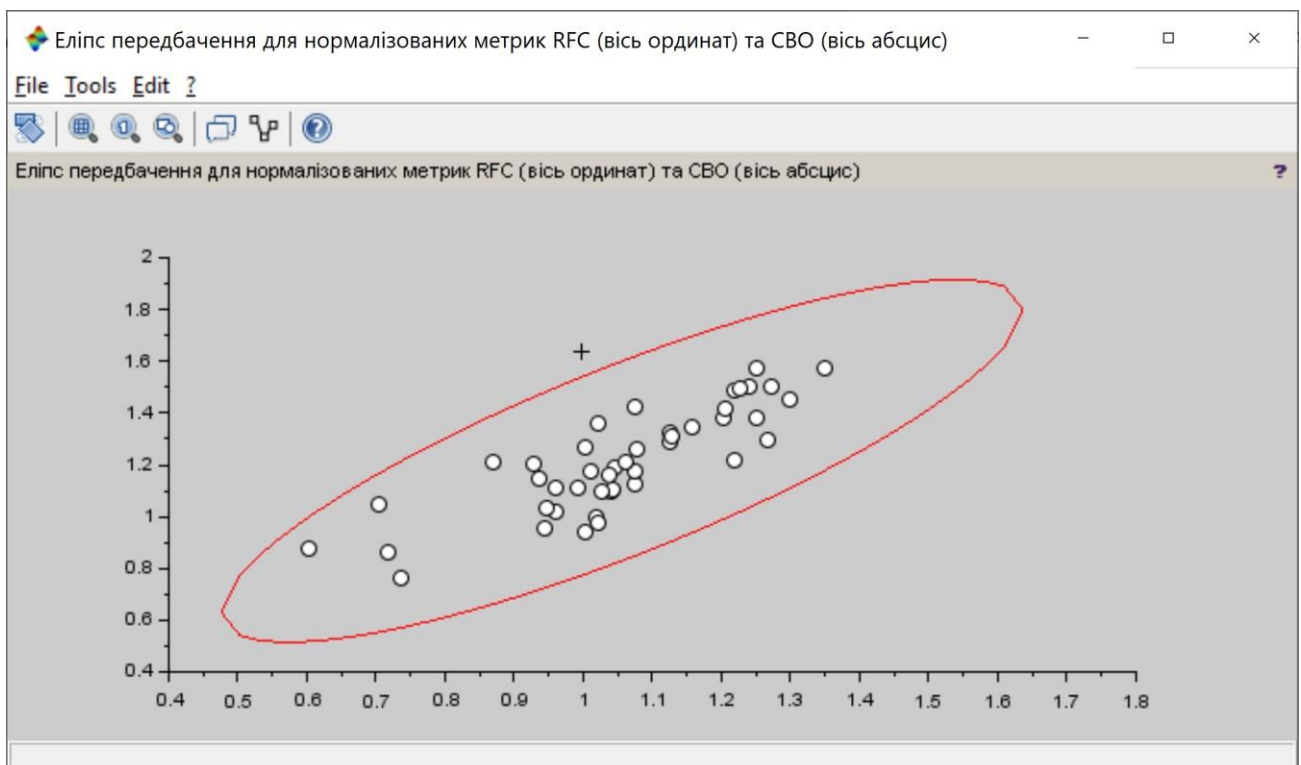


Рисунок 3.8. – Вихід точки з координатами, які відповідають значенням нормалізованих метрик RFC та CBO застосунку Metro\_systems, за еліпс передбачення

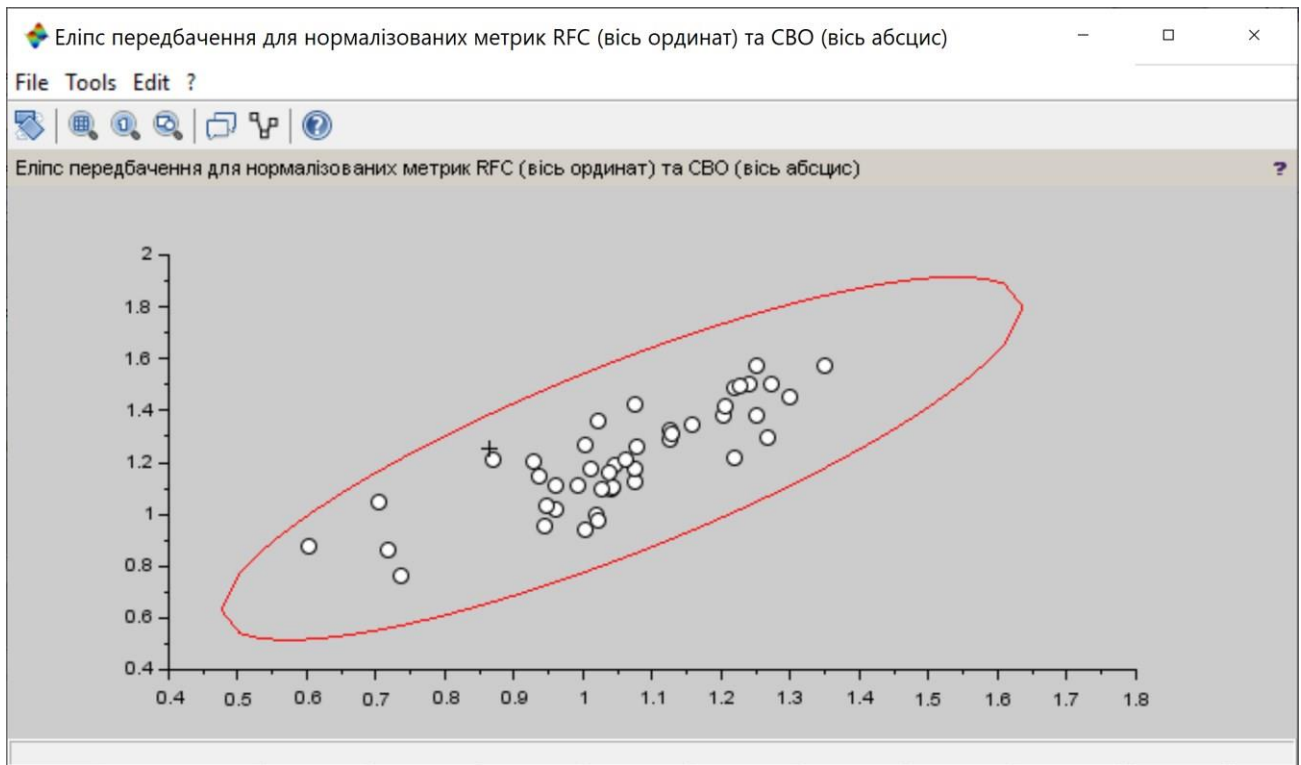


Рисунок 3.9. — Точка з координатами, які відповідають значенням нормалізованих метрик RFC та CBO застосунку TuxGuitar, що знаходиться всередині еліпсу передбачення

На рис. 3.9 зображено у вигляді хреста точка з координатами, які відповідають значенням нормалізованих метрик RFC та CBO застосунку TuxGuitar, що знаходиться всередині еліпсу передбачення (червона лінія). Тобто застосунок TuxGuitar не має великої складності ООП через зв'язки між класами у порівнянні з тими застосунками, для яких було побудовано зазначений еліпс передбачення.

Отримані результати практично збігаються з результатами, що наведені у табл.2.1, та оцінками інших авторів. Це свідчить про працездатність розробленої програми.

### 3.5 Висновки за розділом 3

У даному розділі виконано постановку задачі на розробку проекту програми для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java, розроблено ескізний, технічний та робочий проекти програми для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java.

1) В ескізному проекті було виявлено варіанти використання та побудовано їх сценарії, розроблено зовнішню специфікацію програми для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java та стани цієї програми при виконанні.

2) В технічному проекті було спроектовано програму для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java на основі покрокової деталізації зверху вниз, розроблено специфікації модулів програми, побудовано алгоритми модулів програми.

3) В робочому проекті було розроблено програму для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java, здійснено тестування програми, розроблено програмну документацію, здійснено випробування програми. При випробуванні програми також застосовувалися дані метрик двох відомих застосунків з відкритим кодом на Java – це TuxGuitar та Metro\_systems. Результати випробування програми

практично збігаються з результатами, що отримані іншими авторами. Це свідчить про працездатність розробленої програми.

## **4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВПРОВАДЖЕННЯ ПРОГРАМИ**

Найбільш важливим моментом при проектуванні комп'ютерної програми для розробника, з економічної точки зору, є процес формування її вартості. Вочевидь, що комп'ютерна програма являє собою дуже специфічний товар з безліччю особливостей.

Створення програми вимагає одноразових витрат на її розробку, придбання необхідних технічних засобів і поточних витрат на функціонування продукту. Економія від функціонування програми визначається з урахуванням витрат на її експлуатацію. Відношення цієї економії до витрат на створення програми характеризує економічну ефективність капітальних вкладень. Економічні показники визначаються по діючим на момент розрахунку оптовим цінам, тарифам і ставкам заробітної плати.

### **4.1 Розрахунок витрат на створення й експлуатацію програми**

Витрати на розробку програми складаються з витрат на зарплату розробника, на амортизацію комп'ютера, на якому виконується розробка, на експлуатацію цього комп'ютера, на засоби розробки та витрат на матеріали і комплектуючі.

Розробка програми виконується програмістом, місячний оклад якого складає 30000 грн. Додаткова заробітна плата складає 20% від основної. Виходячи з цього, основна і додаткова заробітна плата програміста

36000 грн/мес, а вартість сучасного ноутбука складає 26076 грн. (приведена вартість ноутбука Lenovo IdeaPad 3 15ALC6 (82KU00NERA) Arctic Grey: Екран 15.6" TN (1920x1080) Full HD, матовий / AMD Ryzen 3 5300U (2.6 - 3.8 ГГц) / RAM 8 ГБ / SSD 128 ГБ / AMD Radeon Graphics / без ОД / Wi-Fi / Bluetooth / веб-камера / Windows 11 Професійна на 1ПК (ESD - електронна ліцензія в конверті, всі мови) (FQC-10572)). При вартості кіловат-години електроенергії у 1,7166 грн. (II клас напруги до 27,5 кВт, разом з ПДВ), розраховується вартість розробки ПЗ. Витрати на допоміжні матеріали приведені в табл. 4.1

Таблиця 4.1 – Витрати на допоміжні матеріали

| Пункти витрат                     | Сума   |
|-----------------------------------|--------|
| Папір                             | 180,00 |
| Заправлення картриджа до принтера | 180,00 |
| CD-диск                           | 12,00  |
| Непередбачені витрати             | 300,00 |
| Разом матеріали і комплектуючі    | 672,00 |

Вартість програми розраховуємо за формулою:

$$\text{Спр} = (\text{Ззп} + \text{Зсз} + \text{Ззг} + \text{Зе}) * \text{T} + \text{Зм}, \quad (4.1)$$

де Т – тривалість розробки, міс.; Ззп – основна і додаткова заробітна плата обслуговуючого персоналу, грн.; Зсз – відрахування на соціальні заходи або єдиний соціальний внесок (36,76% від основної і додаткової заробітної плати), грн.; Ззг – загальногосподарські витрати (10% від основної заробітної плати), грн.; Зе – витрати на електроенергію; Зм – витрати на основні і допоміжні матеріали. Зе при споживанні потужності 0,5 кВт, тривалості роботи за місяць, рівної  $22 * 8 = 176$  годин, вартості кіловат-години електроенергії 1,7166 грн. складає



$$З_е = 176 * 1,7166 * 0,5 = 151,06 \text{ грн.}$$

Таблиця 4.2 – Витрати на розробку ПЗ

| Найменування витрат              | Одиниця | Кількість |
|----------------------------------|---------|-----------|
| Тривалість розробки              | Мес     | 1,5       |
| Основна і додаткова ЗП           | Грн.    | 36000     |
| Відрахування на соціальні заходи | Грн.    | 13233,6   |
| Загальногосподарські витрати     | Грн.    | 500       |
| Витрати на матеріали             | Грн.    | 672,00    |
| Витрати на електроенергію        | Грн.    | 151,06    |

Відповідно до формули (4.1) вартість програми:

$$С_{пр} = (36000 + 13233,6 + 500 + 151,06) * 1,5 + 672,00 = 75498,99 \text{ грн.}$$

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

$$А_{об} = 26076 * 0,6 = 15645,6 \text{ грн.}$$

У масштабах підприємства річні витрати на основні і допоміжні матеріали (гнучкі диски, папір) визначаються в розмірі 5% вартості основного устаткування:

$$В_м = 16800 * 0,05 = 1303,8 \text{ грн.}$$

Річний обсяг робіт ПК у годинах визначається в такий спосіб:

$$\Phi_м = 264,5 * T_з, \quad (4.2)$$

де  $T_з$  – це середнє місячне завантаження устаткування (близько 4 годин); 264,5 – середня кількість робочих днів у році.

Отже, річний обсяг роботи ПК складе:

$$\Phi_m = 264,5 \cdot 4 = 1058 \text{ годин.}$$

Витрати на електроенергію  $З_e$  при 1013 годинах роботи устаткування в рік складуть

$$З_e = 1058 \cdot 0,5 \cdot 1,7166 = 908,08 \text{ грн.}$$

Експлуатаційні витрати для ПК за рік складуть:

$$З_{зр} = 15645,6 + 1303,8 + 908,08 = 17857,48 \text{ грн.}$$

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

$$З_{се} = 75498,99 + 17857,48 = 93356,47 \text{ грн.}$$

#### 4.2 Економічна ефективність розробки і впровадження програми

Основним показником економічної ефективності функціонування програми є підвищення ефективності керування інформацією у вигляді зниження витрат на керування при одночасному збільшенні швидкості і якості одержання потрібного результату.

Крім багатьох інших негативних ефектів, ручна обробка інформації спричиняє наступні негативні економічні ефекти:

- високі витрати на складування паперових документів (сейфи, шафи, папки й ін.);
- підвищені витрати на канцтовари;

– витрати, пов'язані з роботою виявлення раніше допущених помилок (людський фактор).

До числа основних факторів, що визначають приріст прибутку в зв'язку з упровадженням програми, відносяться:

- підвищення продуктивності праці;
- вивільнення робочого часу.

Крім того, не піддається прямій грошовій оцінці підвищення оперативності керування, якість одержуваних результатів, поліпшення організації праці і т.д.

Обов'язковою умовою визначення економічної ефективності програми є порівнянність усіх показників у часі, за цінами й іншими нормами, використовуваними для визначення показників, за змістом і колом елементів витрат.

Визначимо пряму економічну ефективність, ґрунтуючись на тому, що впровадження програмного продукту вивільняє 0,6 працівника (за експертною оцінкою фахівців підприємства).

Зарплата 0,6 працівника в рік складає:

$$(30000 \cdot 12) \cdot 0,6 = 216000 \text{ грн.}$$

Річний економічний ефект розраховується по формулі:

$$E_p = \Delta C_n - E_n \cdot k,$$

де  $\Delta C_n$  – вивільнені кошти після впровадження програми (216000 грн.) мінус експлуатаційні витрати (17857,48 грн.);  $E_n$  – коефіцієнт ефективності (дорівнює коефіцієнту амортизації – 0,6);  $k$  – одноразові витрати на впровадження програми (75498,99 грн.).

Річний економічний ефект буде

$$E_p = 198142,52 - 45299,39 = 152843,13 \text{ грн.}$$

Строк окупності програми розраховується по формулі:

$$T = \frac{k}{\Delta C_n} = \frac{75498,99}{198142,52} = 0,38 \text{ (року).}$$

Отже строк окупності програми складає приблизно 4,6 місяця.

## 5 ОХОРОНА ПРАЦІ

### 5.1 Аналіз існуючої проблеми охорони праці

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності (ст. 1 Закону України «Про охорону праці» [20]). Як правовий інститут охорона праці – це чималий комплекс правових норм. Можна визначити охорону праці також як систему забезпечення безпеки, збереження здоров'я і працездатності людини в процесі праці, засновану на сукупності законодавчих актів і відповідних їм соціально-економічних, технічних, гігієнічних і організаційних заходів.

Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці.

Власник або уповноважений ним орган повинен впроваджувати сучасні засоби техніки безпеки, які запобігають виробничому травматизму, і забезпечувати санітарно-гігієнічні умови, що запобігають виникненню професійних захворювань працівників.

На власника або уповноважений ним орган покладається систематичне проведення інструктажу (навчання) працівників з питань охорони праці, протипожежної охорони.

Охорона праці – один з центральних інститутів трудового права. Він має виключно практичне значення. Недодержання вимог охорони праці створює небезпеку для здоров'я і життя працівників. У свою чергу і ті, кого законодавець називає власником або уповноваженим ним органом, несуть сувору, у тому числі й кримінальну відповідальність за порушення правил охорони праці.

## 5.2 Аналіз шкідливих та небезпечних факторів у офісному приміщенні з персональними комп'ютерами

Небезпечним називається виробничий фактор, вплив якого на працюючу людину в певних умовах приведе до травми або іншому раптовому різкому погіршенню здоров'я. Якщо ж виробничий фактор приведе до захворювання або зниження працездатності, то його вважають шкідливим. Залежно від рівня й тривалості впливу шкідливий виробничий фактор може стати небезпечним.

На офісного працівника при роботі впливають дві основні групи шкідливих факторів:

1) Фізичні, до яких відносять фактори, що породжуються безпосереднім впливом оточуючого середовища (електромагнітне випромінювання, освітленість робочого місця, температура повітря, шум).

2) Психофізіологічні, до яких відносять фізичне та нервово-психічне перевантаження організму (зорове напруження, статичні навантаження, нервово-психічне навантаження).

Освітлення – отримання, розподіл та використання світлової енергії для забезпечення нормальних умов праці. Світло, крім забезпечення фізичної можливості розрізняти предмети праці, також впливає на психічний стан людини та перебіг фізіологічних процесів в організмі. Раціонально влаштоване освітлення виробничих приміщень позитивно впливає на працівників, підвищує ефективність та безпеку праці, знижує втому та травматизм, забезпечує високу працездатність.

За вимогами санітарних норм освітлення повинно бути достатньо яскравим, рівномірним, не засліплювати очі та не створювати відблисків на робочій поверхні. За спектральним складом освітлення має наближатися до сонячного світла. Гігієнічними нормами вимагається максимально використовувати природне освітлення, оскільки денне світло краще сприймається органами зору. Штучне освітлення у виробничих та адміністративно-громадських приміщеннях має здійснюватись системою загального рівномірного освітлення, допускають застосування системи комбінованого освітлення. Значення освітленості на поверхні робочого столу в зоні розміщення документів має становити 300-500лк. Як джерела світла для штучного освітлення мають застосовувати переважно люмінесцентні лампи типу ЛБ.

Одним з найнесприятливіших факторів, що зменшує працездатність та уважність робітників, створює передумови до травматизму та захворювань, є шум. Інтенсивний шум та вібрації негативно впливають не лише на слух, а і на

нервову, серцево-судинну, та більшість інших систем організму, викликаючи зниження слуху, гіпертонію, психічні розлади тощо.

Гігієнічними нормативами (СН 3222-85, ГОСТ 12.1.003-85, ГР 2411-81) встановлені допустимі рівні звуку різних частот для робітників певних професій. У таблиці 5.1 наведені нормативи для людей, що працюють з ЕОМ.

Таблиця 5.1 – Допустимі рівні шуму

| Вид трудової діяльності                                                       | Рівні звукового тиску в дБ октавних смугах із середньо геометричними частотами, ГЦ |    |     |     |     |      |      |      |      |                                                   |
|-------------------------------------------------------------------------------|------------------------------------------------------------------------------------|----|-----|-----|-----|------|------|------|------|---------------------------------------------------|
|                                                                               | 31,5                                                                               | 63 | 125 | 250 | 500 | 1000 | 2000 | 4000 | 8000 | Рівні звуку, еквівалентні рівні звуку, дБа/дБАекв |
| Програмісти ЕОМ                                                               | 86                                                                                 | 71 | 61  | 54  | 49  | 45   | 42   | 40   | 38   | 50                                                |
| Оператори в залах обробки інформації на ЕОМ ті оператори комп'ютерного набору | 96                                                                                 | 83 | 74  | 68  | 63  | 60   | 57   | 55   | 54   | 65                                                |
| В приміщеннях для розташування шумних агрегатів ЕОМ                           | 103                                                                                | 91 | 83  | 77  | 73  | 70   | 68   | 66   | 64   | 75                                                |



Виходячи з даних норм, при проведенні комплексу заходів зі зменшення шуму усувається не будь-яка віброакустична активність обладнання, а тільки та, яка перевищує встановлений граничний рівень та шкідливо впливає на організм людини.

Усунення промислового шуму в першу чергу відбувається шляхом вдосконалення технологічних процесів та обладнання. При цьому в першу чергу усуваються найбільш сильні джерела шуму. Також необхідно знешумлювати усе обладнання, яке цього потребує, оскільки знешумлення окремих одиниць при наявності великої кількості обладнання, що продовжує випромінювати шум, недоцільно.

Самопочуття та працездатність людини також сильно залежать від теплового стану організму, на який впливають такі фізичні фактори виробничого середовища, як температура повітря, вологість, швидкість руху повітря, атмосферний тиск, доля кисню у повітрі тощо. Наприклад, якщо кількість кисню в повітрі зменшиться до 12 % то утруднюється дихання. В таких умовах людина напружує дихальний апарат, дихає частіше; такий стан людина витримує до 0,5 години. Перегрівання організму відбувається за умов надлишкового конвективного випромінювання тепла нагрітих поверхонь. При перегріванні вступають в активну реакцію пристосувальні функції організму. При цьому активізується робота серцево-судинної та дихальної систем, відбувається інтенсивне потовиділення, котре сягає 5 л за зміну. З потом втрачається велика кількість мінеральних солей та вітамінів. Вологість повітря суттєво впливає на терморегуляцію людського організму. Підвищення відносної вологості повітря у виробничому приміщенні ускладнює терморегуляцію, зменшення тепловиділення організмом. Фізіологічно оптимальною є відносна вологість в межах 40..60 %.

Сукупність описаних показників стану навколишнього середовища називають мікрокліматом. Для приміщень з ЕОМ встановлені норми мікроклімату приведені у таблиці 5.2.

Робота комп'ютерів і периферійних пристроїв можлива завдяки електричному струмові, до джерел якого вони підключаються, тому існує небезпека ураження користувача струмом. Щоб уникнути нещасних випадків варто строго дотримуватися правил електробезпеки.

Таблиця 5.2 – Норми мікроклімату для приміщень з ЕОМ

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

Небезпека електричного струму на відміну від інших небезпек збільшується тим, що людина не в змозі без спеціальних приладів виявити напругу дистанційно, а також швидкоплинністю поразки – небезпека виявляється, коли людина вже уражена. Ураження струмом приводить до різних порушень в організмі, викликаючи як місцеву поразку тканин і органів, так і загальне ураження організму.

Людина відчуває дратівну дію змінного струму промислової частоти силою 0,6 -1,5 мА і постійного струму 5-7 мА. Ці струми не представляють серйозної небезпеки для організму людини, тому що при такій силі струму можливе самостійне звільнення людини від контакту зі струмоведучими частинами. Для перемінного струму промислової частоти сила невідпускаючого струму знаходиться в межах 6-20 мА. Постійний струм не викликає невідпускаючого ефекту, але приводить до сильних болючих відчуттів, сила такого струму 15-80 мА і більше.

### 5.3 Розрахунок системи штучного освітлення у офісному приміщенні з персональними комп'ютерами

У приміщенні необхідно організувати змішане освітлення, тобто сполучення природного і штучного освітлення. Природне освітлення створюється бічним освітленням через вікно. Штучне освітлення використовується при недостатньому природному освітленні. У даному приміщенні використовується загальне штучне освітлення.

Розрахунок його здійснюється по методу світлового потоку з урахуванням потоку, відбитого від стін і стелі.

Потрібно зробити розрахунок загального освітлення для офісного приміщення. Довжина  $A=4,5$  м, ширина  $B=3,0$  м, висота  $H=2,5$  м. Висота робочої поверхні  $h_p = 1,0$  м. Для освітлення використовується світильник УПМ-15. Мінімальна освітленість лампи накаливання по нормах  $E_{\min}=75$ лк.

Коефіцієнт відображення стелі  $S_p=70\%$ , стін  $S_c=50\%$ , робочої поверхні  $S_r=10\%$ .

Напруга мережі 220 В.

Відстань від стелі до робочої поверхні:

$$H_o = H - h_p = 2,5 - 1 = 1,5 \text{ м.}$$

Відстань від стелі до світильника:

$$h_c = 0,2 * H_o = 0,2 * 1,5 = 0,3 \text{ м.}$$

Висота підвісу світильника над освітленою поверхнею:

$$h = H_o - h_c = 1,5 - 0,3 = 1,2 \text{ м.}$$

Висота підвісу світильника над підлогою:

$$H_n = h + h_p = 1,2 + 1 = 2,2 \text{ м.}$$

Для досягнення найбільшої рівномірності освітлення приймаємо відношення  $L/h = 1,5$ .

Відстань між центрами світильників:

$$L = 1,5 * h = 1,5 * 1,2 = 1,8 \text{ м.}$$

Необхідна кількість світильників:

$$N = S/L^2 = (4,5 * 3) / (1,8)^2 = 4,2 \approx 4.$$

Приймаємо кількість світильників: 4.

Індекс приміщення:

$$i = (A * B) / (h * (A + B)) = (4,5 * 3) / (1,2 * (4,5 + 3)) = 1,5.$$

Коефіцієнт використання світлового потоку  $\eta = 0,51$ .

Світловий потік однієї лампи:

$$\begin{aligned} \Phi &= (E_{\min} * S * K_3 * Z) / (N * \eta) = \\ &= (75 * 13,5 * 1,3 * 1,15) / (4 * 0,51) = 741 \text{ лм.} \end{aligned}$$

За знайденим світловим потоком з таблиці “Параметри ламп розжарювання загального призначення з розрахунковими напругами 130 и 220 В” обираємо лампу Б215-225-75 потужністю 75 Вт, що має світловий потік 950 лм, найбільш близький до розрахункового.

Фактична освітленість:

$$E_f = E_{\min} * \Phi_{\text{л}} / \Phi_{\text{р}} = 75 * 950 / 741 = 96,15 \text{ лк.}$$

Загальна потужність:

$$P_{\text{общ}} = P_{\text{л}} * N = 75 * 4 = 300 \text{ Вт} = 0,3 \text{ кВт.}$$

Таким чином, розрахунок загального освітлення для офісного приміщення можна вважати виконаним, а його результати такими, що задовольняють загальноприйнятим нормам промислового освітлення.

#### 5.4 Розробка заходів щодо зменшення впливу шкідливих та небезпечних факторів

Сьогодні фахівці в області ергономіки вже зрозуміли, що не можна знайти ідеальне положення, у якому можна перебувати і працювати протягом усього робочого дня. Для більшості людей комфортабельним робочим місцем повинне бути таке, котре можна пристосувати не менш чим для двох позицій, при цьому положення крісла, монітора повинні щораз відповідати виконуваний роботі і звичкам. Багато хто вважають, що для роботи на комп'ютері більше підходять вертикальне і злегка похиле положення. Можливо, щоб крісло було злегка нахилене вперед.

Схеми розміщення робочих місць із персональним комп'ютером повинні ураховувати відстані між робочими столами з відеомоніторами (в напрямку тилу поверхні одного відеомонітора й екрана іншого відеомонітора), які повинні бути не менш 2 м, а відстань між бічними поверхнями відеомоніторів – не менш 1,2 м.

Робочий стіл повинен мати простір для ніг висотою не менш 600 мм, шириною не менш 500 мм, глибиною на рівні колін не менш 450 мм і на рівні витягнутих ніг не менш 650 мм.

Дисплей. Положення тіла звичайно відповідає напрямку погляду; дисплеї, розташовані занадто низько чи під неправильним кутом, є основними причинами сутулості. Відстань від дисплея до очей може варіюватися в залежності від характеру виконуваної роботи – 40-70 см. При роботі з текстом відстань від екрана до очей повинна лише небагато перевищувати відстань між книгою й очима і складати 40-45 см. Необхідно давати відпочинок очам і час від часу їх просто закривати. Не можна допускати, щоб очі увесь час були сфокусовані на одній відстані, працюючи з текстом, рекомендується установлювати великий шрифт. Дуже важливо, щоб екран монітора не мерехтів. Частота регенерації зображення повинна бути не менш 72 Гц, тому що при більш низькій частоті помітне мерехтіння, хоча люди з особливо чутливим зором можуть помітити його і при більш високій частоті. Їм доведеться підібрати графічну плату і монітор з частотою регенерації 85 Гц і вище.

Необхідно уникати того, щоб монітор був звернений убік вікна, оскільки інтенсивна освітленість області зору може затопити потоками світла очі і розмити зображення на сітківці. Якщо приходить сидіти поруч з вікном, то треба розташуватися під прямим кутом до нього, причому екран дисплея повинний бути перпендикулярний шибці – цим виключаються відблиски на екрані.

Для зниження впливу низькочастотного електромагнітного випромінювання монітора на ЕЛТ рекомендується вибирати монітор, що задовольняє стандарту MPR II, чи TCO 92-95. У протилежному випадку необхідно установити на екран захисний фільтр, найбільш сучасні моделі, затримують до 99 % електромагнітного випромінювання.

Форма спинки крісла повинна повторювати форму спини працюючого. Крісло необхідно установити на такій висоті, щоб не почувався тиск на куприк

(крісло розташоване занадто низько) на стегна (крісло розташоване занадто високо). Фахівці з ергономіки, вважали, що кут між стегнами і хребтом повинний складати 90°, однак недавно проведені дослідження показали, що більшість людей переважно сидять трохи відкинувшись.

Клавіатуру необхідно установити так, щоб не треба було до неї тягтися і нахилитися вперед. При зміні положення тіла (наприклад, з вертикального на похиле) обов'язково треба перемінити положення клавіатури і дисплея. Може виявитися корисною регульована підставка клавіатури, завдяки якій можна без усякої напруги працювати з маніпулятором "миша". Можна поставити клавіатуру і на коліна. Руки при роботі повинні бути прямими в зап'ястях і зігнуті в ліктях приблизно під прямим кутом. Пальці також повинні бути злегка зігнуті. Удари по клавішах не повинні бути занадто сильними.

Зручна висота столу особливо важлива в тому випадку, коли на ньому розташовується клавіатура. Якщо в клавіатури немає підставки, а висоту столу не можна змінити (і він занадто високий), то треба вище підняти сидіння крісла, а під ноги підставити лавочку: чи що-небудь інше. Якщо стіл занадто низький, необхідно підкласти що-небудь під його ніжки.

Щогодини необхідно робити перерву в роботі. У цей час рекомендується робити вправи для зап'ясть. Нижче описаний можливий комплекс вправ.

- Покласти руку на край столу долонею вниз. Взявшись, за пальці, іншою рукою відвести кисть назад і утримувати протягом 5 секунд. Повторити для іншої руки.

- Злегка упертися рукою в стіл і на 5 секунд напружити пальці в зап'ястя. Повторити іншою рукою.

- Сильно зжати пальці, а потім розпрямити їх.



Виконання приведених рекомендацій дозволяє скоротити вплив шкідливих виробничих факторів на організм людини.

## **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.

Відносини у галузі охорони навколишнього природного середовища в Україні регулюються Законом України № 1268-ХІІ від 26.06.91 р. «Про охорону навколишнього природного середовища» [21], а також розроблюваними відповідно до нього земельним, водним, лісовим законодавством, законодавством про надра, про охорону атмосферного повітря, про охорону і використання рослинного і тваринного світу та іншим спеціальним законодавством.

Цей Закон 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.

Охорона навколишнього природного середовища є системою державних і суспільних заходів, направлених на збереження, відтворювання і раціональне використання природних ресурсів і поліпшення полягання природного середовища, і є частиною прикладної екології.

В міру прискорення темпів науково-технічного прогресу дія людей на природу стає все більш могутньою. І в даний час воно вже сумирно із дією природних чинників, що приводить до якісної зміни співвідношення сил між суспільством і природою. На сучасному етапі людство поставлено перед чинником виникнення в природі незворотних процесів, нових шляхів переміщення і перетворення енергії і речовини. В природу втручається все більше і більше нових речовин, чужих їй, часом сильно токсичних для організмів. Частина з них не включається в природний круговорот і нагромаджується в біосфері, що приводить до небажаних екологічних наслідків.

Накопичення промислових відходів сприяє підвищенню захворюваності людей і тварин, прискоренню корозії машин і металевого устаткування,

зниженню врожайності сільськогосподарських культур і продуктивності тваринництва, прискореному і нераціональному використуванню природних ресурсів і енергії, погіршенню багатьох властивостей екологічних систем, загибелі деяких унікальних природних територіальних комплексів, зникненню окремих видів тварин і рослин.

Основну загрозу для навколишнього середовища в межах галузі розробки ПЗ становлять комплекси електронно-обчислювальних машин (надалі ЕОМ), а саме електромагнітні випромінювання від них.

Дослідження останніх років показали, що електромагнітні випромінювання вищезгаданих електронних пристроїв містять торсіонову компоненту, котра несе інформацію про процеси, що відбуваються в тому чи іншому електронному пристрої.

Торсіонові поля мають високу проникаючу здатність і їх неможливо заекранувати.

Останніми роками при проектуванні (конструюванні), виготовленні (будівництві) і експлуатації технічних і систем управління в різних сферах діяльності надзвичайно широко застосовуються персональні комп'ютери і всілякі комп'ютерні програми.

Робота з комп'ютерами програмістів, операторів і інших користувачів пов'язана з додатковими шкідливими і небезпечними чинниками, що негативно впливають на організм людини. Наприклад, шкідлива дія на зір надає не при тримання стандартних візуальних ергономічних параметрів екрану, розмір мінімального елементу відображення, мерехтіння зображення, відбивна здатність (відблиски) і ін. Працюючий комп'ютер створює електромагнітне поле, що шкідливо діє на організм людини. Це поле може викликати радіоперешкоди, тобто заважати роботі радіо і телеапаратури, що призводить до

зниження надійності технічної системи або системи управління, до збільшення ризику виникнення аварійної ситуації у виробничому середовищі. Для забезпечення безпеки роботи з комп'ютером розроблені і повинні повсюдно застосовуватися стандарти на монітори, вимоги до приміщень для експлуатації комп'ютерів і до організації і устаткування робочих місць.

## 6.2 Утилізація відходів комп'ютерної техніки

Використання комп'ютерів вимагає вирішення таких важливих питань, як утилізація відходів (мікросхеми з вмістом кольорових металів, плати, дискети).

При утилізації старих комп'ютерів відбувається їх розділ на сім фракцій: метали, пластмаси, штекери, дроти, батареї, скло. Жодна деталь не йде для повторного використання, оскільки не можна гарантувати їх надійність, але у формі вторинної сировини вони йдуть на виготовлення нових комп'ютерів або інших пристроїв.

Детально розглянемо декілька прикладів переробки відходів обчислювальної техніки.

Гадолінієво-галлеві гранати (ГГГ) використовуються у виробництві компонентів пристроїв, що запам'ятовують. В ході обробки біля 80 % вихідного матеріалу перетворюється на відходи або відбраковується. ГГГ мають високу вартість і їх виділення з відходів представляє інтерес з економічної точки зору.

При здобутті досить чистих продуктів можливі повторне їх використання як вихідний матеріал. При цьому значно підвищується економічність виробництва заготовок з ГГГ. Під терміном відходи маються на

увазі кристалічні залишки (залишки середовища для зростання кристалів, частини кристалів, виробництва, що утворюються на різних стадіях), а також дрібний порошок, що виходить при різанні, шліфовці, поліровці кристалів граната або подібних матеріалів.

Переробка цих відходів протягом ряду останніх років викликає труднощі і не вирішена до цих пір. Всі попередні спроби були безуспішними із-за низької розчинності цих складних оксидів.

Вдосконалений процес, розроблений Е. Гуссетом (патент США 4-198231 від 15 квітня 1980 року, фірма «Свісс Алюмініум Лтд.», Швейцарія), призначений для виділення галію і гадолінія з відходів, що містять обидва ці елементи у вигляді оксидів або з'єднань, що перекладаються в оксиди. Відходи дрібно подрібнюються і потім розчиняються в сильних мінеральних кислотах. Гадоліній осідає з очищених розчинів у вигляді оксалата, галій виділяється в металевому вигляді електролітично. Електролітичне виділення галію може проводитися до виділення гадолінія у вигляді оксалата з розчину.

Розглянемо приклад проведення такого процесу. Відходи піддаються переробці, є залишки завантаження в пристрої для зростання кристалів, розколені частини кристалів зі всіх стадій переробки, дрібнозернисті порошки і пудри після операції різання, шліфовки і поліровки гранатів  $GdGaO$ . Змельчений порошок після обробки кристалів ГГГ висушується при 1200 С і потім нагрівається при 6000 С протягом декількох годин для розкладання летких забруднень.

Дрібнозернисті відходи в кількості 1000 г розміром менше 40 мкм., що містять 34 % галію і 46 % гадолінія кип'ятяться із зворотним холодильником протягом двох годин в 2100 мл 35 %-ной соляної кислоти. Це відповідає 99 % завантаження. Після кип'ятіння частина відходів, що не розчинилася,

фільтрується і промивається. Після висушування залишок важить 20 г. Залишки такого типу об'єднуються і піддаються повторній обробці кип'яченням. Фільтрат, що включає промивні води, об'ємом 2300 мл, обробляється 4000 г металевого галію при 500 С в течії 45 хвилин. Метал має максимально диспергувати.

В ході цієї операції благородніші елементи, присутні в розчині, виділяються і частково розчиняються в галії до його насичення і далі охолоджуються у вигляді інтерметалевих включень або в елементарній формі. Висадження можна проводити з меншою кількістю галію. Метал може періодично замінюватися на нову порцію до досягнення необхідної міри очищення. В результаті процесу очищення виходить розчин з вмістом галію 140 г/л і гадолінія 190 г/л.

Встановлюється величина  $ph=1$  шляхом додавання 900 мл 4%-ного розчину перекису водню для окислення домішок заліза. Осадження гадолінія проводиться при 500 С шляхом додавання 1500 г кристалічної технічної щавлевої кислоти  $COH \cdot 2HO$ ; суспензія акуратно перемішується 12 годин для повторного осадження у вигляді оксалата гадолінія.

Оксалат гадолінія  $Gd(CO) \cdot 10HO$  відділяється центрифугуванням, промивається 20 мл розбавленої щавлевої кислоти (6 г/л) і висушується при 1300 С; перетворення на оксид гадолінія досягається прожаренням при 8000 С. Подальше очищення може проводитися розчиненням в кислоті і осадженням у вигляді оксалата гадолінія. До рідини після центрифугування і промивань осаду (3300 мл) додають 2300 г КОН при інтенсивному перемішуванні до  $ph = 12$ , 6000 мл розчину з вмістом галію 55 г/л і гадолінія 1 г/л піддається електролізу при 600 Із з використанням катода з нержавіючої сталі і графітового анода при щільності струму близько 0,1 А/кв.см. Після 48 годин осідає 325 г

галію і залишається розчин з вмістом 0,4 г/л, що не піддається подальшій переробці. Обложений метал має чистоту 99,99 % і може бути безпосередньо перетворений в оксид.

При експлуатації ПК витрачаються наступні ресурси:

- електроенергія;
- папір для принтера;
- картриджи з фарбувальною стрічкою.

Для раціонального використання електроенергії не слід залишати включеним комп'ютер і принтер, якщо вони не потрібні в даний час. Друкувати можна з двох сторін. Витрати на папір навряд чи вдасться скоротити удвічі, проте економія буде вельми істотною. Проблему з утилізацією паперових відходів може вирішити вторинна переробка.

Для повторного використання картриджів з фарбувальною стрічкою необхідно купити пристрій для просочення стрічок і тоді картриджи можна буде використовувати 20-40 раз.

Сучасна технологія виготовлення елементів засобів обчислювальної техніки (СВТ) дозволяє досягти дуже низького рівня відмов елементів під час експлуатації. У зв'язку з цим відпадає необхідність проведення ремонтних робіт на місці експлуатації сучасних засобів обчислювальної техніки і як наслідок не утворюються відходи (несправні мікросхеми), що містять дорогоцінні і рідкоземельні метали. Природно, в сервісних центрах, що спеціалізуються на ремонті і технічному обслуговуванні СВТ, має бути організований збір і облік матеріалів, що містять коштовні метали, з подальшою обробкою цих матеріалів на спеціалізованих заводах з метою їх вилучення. У зв'язку з тим, що вітчизняне виробництво сучасних компонентів інформаційних технологій знаходиться в



сьогоднішні дні лише в зачатковому стані, СВТ складаються з парку імпортованих машин і устаткування. Із-за відсутності інформації про вміст дорогоцінних металів в елементах устаткування, строгий облік не представляється можливим і має бути покладений на фахівців експортних фірм. В умовах ринкової економіки підприємства мають бути самі зацікавлені у вторинній переробці, що містять дорогоцінні метали вузлів і елементів за умови неможливості їх використання.

Сумарна маса дорогоцінних металів в ПК, використовуваному в розробці даної дипломної роботи приблизно складає: золото - 0,23 г; срібло - 5,09 г.

Технологічний процес витягання дорогоцінних металів з друкарських плат здійснюється за наступною схемою. Друкарські плати сортуються по переважанню в них кількості дорогоцінних металів, дробляться і подрібнюються, обпалюються і плавляться. В процесі випалення перолітичеському розкладанню піддається пластмасова основа, а основа дорогоцінних металів у вигляді металевих залишків відновлюється до оксидів. Металевий залишок подрібнюється, гранулюється, проходить магнітну сепарацію і відбувається відділення магнітних від немагнітних часток. Отриманий таким чином порошок, розділений по видах дорогоцінних металів, у вигляді гранул розплавляється в індукційних плавильних печах з подальшим розділенням кожного металу окремо.

## ВИСНОВКИ

В кваліфікаційній роботі вирішене наукове завдання з підвищення достовірності оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java за допомогою рівняння еліпсу передбачення для нормалізованих метрик RFC та CBO.

1) Зараз для оцінювання складності ООП через зв'язки між класами застосунків, у тому числі з відкритим кодом на Java, використовуються метрики RFC та CBO. Для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java може бути застосований еліпс передбачення для нормалізованих метрик RFC та CBO. Побудову еліпсу передбачення для нормалізованих метрик RFC та CBO для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java краще здійснювати із застосуванням нормалізуючих перетворень із врахуванням викидів у даних.

2) Перевірку даних для побудови рівняння еліпсу передбачення для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java на наявність викидів виконано за методом, який оснований на нормалізуючих перетвореннях і квадраті відстані Махаланобіса (Mahalanobis Distance). З'ясовано, що за методом, який оснований на нормалізуючих перетвореннях і квадраті відстані Махаланобіса, двовимірні дані містять два двовимірних викиди. Застосування критерія на основі квадрата відстані Махаланобіса до даних без їх нормалізації показало, що дані містять лише один двовимірний викид. Такий результат пояснюється тим, що зазначені дані мають розподіл, який не є гаусівським.

3) Удосконалено рівняння еліпсу передбачення для десяткових логарифмів метрик RFC та CBO для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java на основі нормалізуючого перетворення у формі десяткового логарифму із врахуванням викидів у даних. Зазначене рівняння у порівнянні з існуючими математичними моделями враховує як кореляцію між метриками RFC та CBO, так і те, що їх розподіл не є гаусівським. Це дозволяє підвищити достовірність оцінювання складності об'єктно-орієнтованого проектування через зв'язки між класами застосунків з відкритим кодом на Java.

4) На підставі запропонованої рівняння еліпсу передбачення розроблено програму для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java для системи моделювання Scilab 6.0.0. Була розроблена наступна програмна документація: технічне завдання, текст програми, опис програми, інструкція користувача, програма та методика випробувань, що наведені відповідно у додатках А, Б, В, Г, Д. Були проведені випробування програми згідно з програмою та методикою випробувань, що наведена у додатку Д. За результатами випробувань зроблено висновок про працездатність розробленої програми для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java для системи моделювання Scilab 6.0.0.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Chidamber, S., & Kemerer, C. (1994). A metrics suite for object oriented design. *IEEE Transactions on Software Engineering*, 20(6), 476-493.
2. Coupling Between Object classes (CBO). Access mode: <https://www.cachequality.com/docs/metrics/coupling-between-object-classes-cbo>
3. Response for Class (RFC). Access mode: <https://www.cachequality.com/docs/metrics/response-for-class>
4. Chidamber & Kemerer object-oriented metrics suite. Access mode: <https://www.aivosto.com/project/help/pm-oo-ck.html>
5. Sahraoui, H.A., Godin, R., & Miceli, T. (2000). Can metrics help to bridge the gap between the improvement of OO design quality and its automation? *Software Maintenance : 2000 International Conference (ICSM)* (pp. 154-162). San Jose, CA, USA. DOI: <https://doi.org/10.1109/ICSM.2000.883034>
6. Molnar, A.J., Neamțu, A., & Motogna, S. (2020). Evaluation of software product quality metrics. In E. Damiani, G. Spanoudakis, & L. Maciaszek (eds). *Evaluation of Novel Approaches to Software Engineering : 14<sup>th</sup> International Conference (ENASE 2019), Communications in Computer and Information Science: Vol. 1172* (pp. 163-187). Springer, Cham. DOI: [https://doi.org/10.1007/978-3-030-40223-5\\_8](https://doi.org/10.1007/978-3-030-40223-5_8)
7. Prykhodko S.B. A statistical estimation of the coupling between object metric for open-source apps developed in Java / S. B. Prykhodko, K. S. Prykhodko, T. G. Smykodub // Herald of Advanced Information Technology (Вісник сучасних інформаційних технологій). – 2022. – No. 3 (5). – P. 175-184. DOI: <https://doi.org/10.15276/hait.05.2022.13>

8. Приходько С.Б. Математична модель для оцінювання складності об'єктно-орієнтованого проектування через зв'язки між класами застосунків з відкритим кодом на Java / С. Б. Приходько, **М. В. Резниченко** // Інформаційні технології: моделі, алгоритми, системи (ITMAS – 2022): Матеріали III-ої Всеукраїнської науково-практичної інтернет конференції (26-28 жовтня 2022 р.). – Миколаїв: НУК імені адмірала Макарова, 2022. – С. 6-7. Режим доступу: <http://itconf.nuos.edu.ua/2022/wp-content/uploads/sites/5/NUOS-ITMAS-2022-Proceedings.pdf>

9. Приходько С.Б. Математична модель для перевірки взаємозв'язків між частинами Java-застосунків за метриками RFC та СВО / С. Б. Приходько, Т. Г. Смикодуб // Інформаційні технології: моделі, алгоритми, системи (ITMAS – 2021): Матеріали II-ої Всеукраїнської науково-практичної інтернет конференції (26-28 жовтня 2021 р.). – Миколаїв: НУК імені адмірала Макарова, 2021. – С. 25-26. Режим доступу: <http://itconf.nuos.edu.ua/2021/proceedings/>

10. Prykhodko S. Detecting bivariate outliers on the basis of normalizing transformations for non-Gaussian data / S. Prykhodko, N. Prykhodko, L. Makarova, O. Kudin, T. Smykodub, A. Prykhodko // Advanced Information Systems and Technologies: proceedings of the V international scientific conference, Sumy, May 17-19 2017 / Edited by S. I. Protsenko, V. V. Shendryk. – Sumy: Sumy State University, 2017. – P. 95-97.

11. Mardia K.V. Measures of multivariate skewness and kurtosis with applications / K. V. Mardia // Biometrika. – 1970. – Vol. 57. – P. 519–530. DOI: <https://doi.org/10.1093/biomet/57.3.519>

12. Prykhodko S.B. Constructing the transformed prediction ellipses on the basis of normalizing transformations for bivariate non-Gaussian data /

S. B. Prykhodko, N. V. Prykhodko, O. O. Kudin, T. G. Smykodub // Problems of information technology. – 2017. – № 1 (021). – C.134-138. – ISSN 1998-7005

13. Johnson R.A. Applied Multivariate Statistical Analysis / R. A. Johnson, D. W. Wichern. – Pearson Prentice Hall, 2007. – 800 p.

14. Prykhodko S. Detecting Outliers in Multivariate Non-Gaussian Data on the basis of Normalizing Transformations / S. Prykhodko, N. Prykhodko, L. Makarova, K. Pugachenko // Proceedings of the 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON) «Celebrating 25 Years of IEEE Ukraine Section», Kyiv, Ukraine, May 29 – June 2, 2017. – P. 846-849. DOI: <https://doi.org/10.1109/UKRCON.2017.8100366>

15. Prykhodko S. Application of the Squared Mahalanobis Distance for Detecting Outliers in Multivariate Non-Gaussian Data / S. Prykhodko, N. Prykhodko, L. Makarova, A. Pukhalevych // Proceedings of 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), Lviv-Slavske, Ukraine, February 20–24, 2018. – P. 962-965. DOI: <https://doi.org/10.1109/TCSET.2018.8336353>

16. V. Chew, “Confidence, prediction and tolerance regions for the multivariate normal distribution,” Journal of the American Statistical Association, vol. 61, issue 315, 1966, pp. 605-617. <https://doi.org/10.2307/2282774>

17. I. Olkin, and A.R. Sampson, “Multivariate Analysis: Overview,” in N.J. Smelser, P.B. Baltes, Eds. International encyclopedia of social & behavioral sciences. 1st edn. Elsevier, Pergamon, 2001, pp. 10240-10247.

18. A.A. Afifi, and S.P. Azen, Statistical analysis: a computer-oriented approach. New York; London: Academic Press, 1972.

19. Sparsh6496/Metro\_systems. Access mode: [https://github.com/Sparsh6496/Metro\\_systems](https://github.com/Sparsh6496/Metro_systems)

20. Закон України «Про охорону праці». Режим доступу:  
<https://zakon.rada.gov.ua/laws/show/2694-12#Text>

21. Законом України № 1268-ХІІ від 26.06.91 р. «Про охорону  
навколишнього природного середовища» (із змінами). Режим доступу:  
<https://zakon.rada.gov.ua/laws/show/1264-12#Text>

## **Додаток А – Технічне завдання на розробку програми для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java**

### **Вступ**

Назва розробки – “Програма для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java”, далі програма.

### **Підстави для розробки**

Завдання на кваліфікаційну роботу.

### **Призначення розробки:**

Програма призначена для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java.

### **Вимоги до програми:**

*Вимоги функціональних характеристик.*

Вимоги до складу функцій, що виконуються.

Програма має виконувати такі функції:

Вводити з текстового файлу емпіричні дані з метрик CBO та RFC застосунків з відкритим кодом на Java.

2) Вводити з інтерфейсу програми довірчу ймовірність, метрики CBO та RFC, за якими потрібно оцінити складність ООП застосунку через зв'язки між класами із заданою ймовірністю.

3) Виконувати нормалізацію емпіричних даних за нормалізуючим перетворенням у формі десяткового логарифму.

4) Здійснювати побудову рівняння еліпсу передбачення для нормалізованих метрик CBO та RFC.

5) Обчислювати значення тестової статистики для квадрата відстані Махаланобіса.



7) Здійснювати оцінювання складності ООП застосунку через зв'язки між класами із заданою ймовірністю.

8) Виводити на екран значення квадрата відстані Махаланобіса для нормалізованих метрик CBO та RFC, значення тестової статистики для квадрата відстані Махаланобіса, результати оцінювання складності ООП застосунку через зв'язки між класами із заданою ймовірністю.

9) Виводити у текстовий файл значення квадрата відстані Махаланобіса для нормалізованих метрик CBO та RFC, значення тестової статистики для квадрата відстані Махаланобіса, результати оцінювання складності ООП застосунку через зв'язки між класами із заданою ймовірністю.

*Вимоги до організації вхідних та вихідних даних*

*Вхідні дані:*

- емпіричні дані з метрик CBO та RFC застосунків з відкритим кодом на Java;

- значення довірчої ймовірності, значення метрик CBO та RFC, за якими потрібно оцінити складність ООП застосунку через зв'язки між класами із заданою ймовірністю.

*Вихідні дані:*

- значення квадрата відстані Махаланобіса для нормалізованих метрик CBO та RFC;

- значення тестової статистики для квадрата відстані Махаланобіса.

*Вимоги до надійності не висуваються.*

*Вимоги до умов експлуатації не висуваються.*

*Вимоги до складу й параметрів технічних засобів.*

Програма має експлуатуватись на обладнанні з параметрами не нижчими за наступні: процесор Pentium IV (або еквівалентний) з інструкціями SSE2,

оперативна пам'ять 2 Гб (1 Гб мінімум), простір на жорсткому диску 600 Мб мінімум.

*Вимоги до інформаційної та програмної сумісності.*

Для запуску та роботи з програмним забезпеченням необхідна операційна система Windows Vista/7/8/10 (32 або 64 біти), система моделювання Scilab 6.0.0., компілятор C (Visual Studio 2010 або Visual Express 2010) для компіляції зовнішніх модулів C або C++ та використання моделі в Xcos.

### **Вимоги до програмної документації.**

Програмна документація має включати: технічне завдання на розробку програми, текст програми, опис програми, інструкція користувача, програма та методика випробувань.

**Техніко-економічні показники** не розраховуються.

### **Стадії та етапи розробки.**

Стадії та етапи розробки програми представлені у таблиці.

Таблиця – Стадії та етапи розробки

| Стадія розробки      | Етапи робіт                                  | Термін виконання |
|----------------------|----------------------------------------------|------------------|
| 1. Технічне завдання | Розробка та затвердження технічного завдання | 11.10.2022       |
| 2. Ескізний проект   | Розробка ескізного проекту                   | 13.10.2022       |
|                      | Затвердження ескізного проекту               | 14.10.2022       |
| 3. Технічний проект  | Розробка технічного проекту                  | 20.10.2022       |
|                      | Затвердження технічного проекту              | 21.10.2022       |
| 4. Робочий проект    | Розробка програми                            | 04.11.2022       |
|                      | Розробка програмної документації             | 10.11.2022       |
|                      | Випробування програми                        | 11.11.2022       |

### **Порядок контролю і приймання**

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

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

За результатами приймання програми представниками замовника складається акт приймання програми.

## Додаток Б – Текст програми

```

//Оцінювання складності ООП через зв'язки між класами застосунків з
відкритим кодом на Java
f=figure();//Создание графического окна.
//Устанавливаем размеры окна.
set(f,'position',[0,0,700,300])
//Устанавливаем заголовок окна.
set(f,'figure_name',...
'Oцінювання складності ООП через зв'язки між класами застосунків з
відкритим кодом на Java');

//plot2d(x,y,-9);

lab_text0=uicontrol(f,'style','text','string',...
'Файл з даними','position',...
[350,270,80,20]);
edit_text=uicontrol(f,'style','edit','string',...
'D:\Data_F\CBO_RFC_44.txt','position',[320,240,140,20]);

BtDatInp=uicontrol('style','pushbutton','string','Ввести',...
'CallBack','[x,f1,n]=DataInput()','position',[340,210,100,20]);

lab_text2=uicontrol(f,'style','text','string',...
'Наявність нормалізації','position',...
[500,270,190,20]);
lab_text3=uicontrol(f,'style','text','string',...
'Дані для оцінювання','position',...
[60,270,110,20]);
lab_text4=uicontrol(f,'style','text','string',...
'Значення СВО на рівні застосунку','position',...
[20,240,200,20]);
edit_text4=uicontrol(f,'style','edit','string',...
'4.0','position',[230,240,60,20]);
lab_text5=uicontrol(f,'style','text','string',...
'Значення RFC на рівні застосунку','position',...
[20,210,200,20]);
edit_text5=uicontrol(f,'style','edit','string',...
'7.5','position',[230,210,60,20]);
lab_text6=uicontrol(f,'style','text','string',...
'Довірча ймовірність','position',...
[20,180,200,20]);
edit_text6=uicontrol(f,'style','edit','string',...
'0.995','position',[230,180,60,20]);
//lab_text7=uicontrol(f,'style','text','string',...
//'Довірча ймовірність','position',...
//[20,150,200,20]);
//edit_text7=uicontrol(f,'style','edit','string',...

```

```

//'0.95','position',[230,150,60,20]);
lab_text8=uicontrol(f,'style','text','string',...
'Reзультати оцінювання','position',...
[50,120,120,20]);
lab_text9=uicontrol(f,'style','text','string',...
'Квадрат відстані Махаланобіса','position',...
[20,90,245,20]);
lab_text10=uicontrol(f,'style','text','string',...
'Тестова статистика на основі F розподілу','position',...
[20,60,245,20]);
//lab_text11=uicontrol(f,'style','text','string',...
//'Верхня границя довір.інтервалу ','position',...
//[20,30,245,20]);
//lab_text12=uicontrol(f,'style','text','string',...
//'Нижня границя інтервалу прогноз.','position',...
//[355,60,260,20]);
//lab_text13=uicontrol(f,'style','text','string',...
//'Верхня границя інтервалу прогноз.','position',...
//[355,30,260,20]);
//
//Текстовое поле, определяющее вывод результатов.
textresult=uicontrol(f,'style','text','string','','position',...
[270,90,70,20]);
textresult2=uicontrol(f,'style','text','string','','position',...
[270,60,70,20]);
//textresult3=uicontrol(f,'style','text','string','','position',...
//[270,30,70,20]);
textresult4=uicontrol(f,'style','text','string','','position',...
[190,120,250,20]);
//textresult5=uicontrol(f,'style','text','string','','position',...
//[620,60,70,20]);
//textresult6=uicontrol(f,'style','text','string','','position',...
//[620,30,70,20]);

//Флажок, отвечающий за выбор типа преобразования.
radio_bikv_x=uicontrol('style','radiobutton','string',...
'є (включити), немає (виключити)','value',1,'position',...
[490,240,200,20]);
//
if (get(radio_bikv_x,'value')==0)

else

end;
//JavaAppComplEval
Btnr_pred_int=uicontrol('style','pushbutton','string','Оцінювання','CallB
ack',...
'JavaAppComplEval(x,n)', 'position',[340,150,100,20]);

```

```

BtClose=uicontrol('style','pushbutton','string','Закрити',...
'CallBack','W_Close()','position',[350,5,100,20]);

function [x, f1, n]=DataInput()
dir_name=get(edit_text,'string');

file_PC=mopen(dir_name,'r');
file_PC2=mopen('D:\Data_F\CBO_RFC_44_data2.txt','w+');
file_PC3=mopen('D:\Data_F\CBO_RFC_44_data3.txt','w+');
i=1;
//n=52;
while (feof(file_PC))<1
    x1(i)=mfscanf(file_PC,'%f');
    x2(i)=mfscanf(file_PC,'%f');
    //    x3(i)=mfscanf(file_PC,'%f');
    i=i+1;
end;
n=i-1;

fclose(file_PC);

for i = 1:n
    //    yx(i,1)=y(i); yx(i,2)=x1(i); yx(i,3)=x2(i); yx(i,2)=x3(i);
    x(i,1)=x1(i); x(i,2)=x2(i);
    //    x(i,3)=x3(i);
end;
//for i=1:n
    fprintf(file_PC2,'%f\t%f\n',x1,x2);
//end;

for i = 1:n
    for j = 1:2 fprintf(file_PC3, "%5.3f ", x(i,j)); end;
    fprintf(file_PC3, "\n");
end;

fclose(file_PC2);
fclose(file_PC3);
f1=figure(); //Создание графического окна.
//Устанавливаем размеры окна.
set(f1,'position',[50,50,700,300]);
//Устанавливаем заголовок окна.
set(f1,'figure_name','Емпіричні дані: СВО (вісь абсцис) та RFC (вісь ординат)');
plot2d(x1,x2,-9);

x1A=0; x2A=0;
//x3A=0;
for i=1:n
    x1A=x1A+x(i,1); x2A=x2A+x(i,2);

```

```

//      x3A=x3A+x(i,3);
end;
x1A=x1A/n; x2A=x2A/n;
//x3A=x3A/n;
Sx1x1=0; Sx2x2=0;
//Sx3x3=0;
Sx1x2=0;
//Sx1x3=0; Sx2x3=0;
for i=1:n
    Sx1x1=Sx1x1+(x(i,1)-x1A)*(x(i,1)-x1A);
    Sx1x2=Sx1x2+(x(i,1)-x1A)*(x(i,2)-x2A);
//    Sx1x3=Sx1x3+(x(i,1)-x1A)*(x(i,3)-x3A);
//    Sx2x3=Sx2x3+(x(i,2)-x2A)*(x(i,3)-x3A);
    Sx2x2=Sx2x2+(x(i,2)-x2A)*(x(i,2)-x2A);
//    Sx3x3=Sx3x3+(x(i,3)-x3A)*(x(i,3)-x3A);
end;
Sxx(1,1)=Sx1x1; Sxx(1,2)=Sx1x2;
//Sxx(1,3)=Sx1x3;
Sxx(2,1)=Sx1x2; Sxx(2,2)=Sx2x2;
//Sxx(2,3)=Sx2x3;
//Sxx(3,1)=Sx1x3; Sxx(3,2)=Sx2x3; Sxx(3,3)=Sx3x3;

Sxx_1=1/Sxx;

z1A=0; z2A=0;
//z3A=0;
for i=1:n
    z1A=z1A+log10(x(i,1)); z2A=z2A+log10(x(i,2));
//    z3A=z3A+log10(x(i,3));
end;
z1A=z1A/n; z2A=z2A/n;
//z3A=z3A/n;
Sz1z1=0; Sz2z2=0;
//Sz3z3=0;
Sz1z2=0;
//Sz1z3=0; Sz2z3=0;
for i=1:n
    Sz1z1=Sz1z1+(log10(x(i,1))-z1A)*(log10(x(i,1))-z1A);
    Sz1z2=Sz1z2+(log10(x(i,1))-z1A)*(log10(x(i,2))-z2A);
//    Sz1z3=Sz1z3+(log10(x(i,1))-z1A)*(log10(x(i,3))-z3A);
//    Sz2z3=Sz2z3+(log10(x(i,2))-z2A)*(log10(x(i,3))-z3A);
    Sz2z2=Sz2z2+(log10(x(i,2))-z2A)*(log10(x(i,2))-z2A);
//    Sz3z3=Sz3z3+(log10(x(i,3))-z3A)*(log10(x(i,3))-z3A);
end;
Szz(1,1)=Sz1z1; Szz(1,2)=Sz1z2;
//Szz(1,3)=Sz1z3;
Szz(2,1)=Sz1z2; Szz(2,2)=Sz2z2;
//Szz(2,3)=Sz2z3;
//Szz(3,1)=Sz1z3; Szz(3,2)=Sz2z3; Szz(3,3)=Sz3z3;

```

```

Szz_1=1/Szz;

zi(1)=log10(x(1,1))-z1A; zi(2)=log10(x(1,2))-z2A;
//zi(3)=log10(x(1,3))-z3A;
ziT=zi';
ziSzz_1=ziT*Szz_1;
result=ziSzz_1*zi;

Df=n-4;
P=0.975;
Q=1-P;

[T]=cdfn("T",Df,P,Q);

endfunction

//Функція оцінювання складності ООП через зв'язки між класами застосунків
з відкритим кодом на Java
function JavaAppComplEval(x, n)
//Считываем значение переменных из текстовых полей и
//преобразовываем их к числовому типу.
x1i=eval(get(edit_text4,'string'));
x2i=eval(get(edit_text5,'string'));
//x3i=eval(get(edit_text6,'string'));
Pconf=eval(get(edit_text6,'string'));

x1A=0; x2A=0;
//x3A=0; yA=0;
for i=1:n
    x1A=x1A+x(i,1); x2A=x2A+x(i,2);
    //    x3A=x3A+x(i,3); yA=yA+yx(i,1);
end;
x1A=x1A/n; x2A=x2A/n;
//x3A=x3A/n; yA=yA/n;
Sx1x1=0; Sx2x2=0;
//Sx3x3=0;
Sx1x2=0;
//Sx1x3=0; Sx2x3=0; SyA=0;
for i=1:n
    Sx1x1=Sx1x1+(x(i,1)-x1A)*(x(i,1)-x1A);
    Sx1x2=Sx1x2+(x(i,1)-x1A)*(x(i,2)-x2A);
    //    Sx1x3=Sx1x3+(x(i,1)-x1A)*(x(i,3)-x3A);
    //    Sx2x3=Sx2x3+(x(i,2)-x2A)*(x(i,3)-x3A);
    Sx2x2=Sx2x2+(x(i,2)-x2A)*(x(i,2)-x2A);
    //    Sx3x3=Sx3x3+(x(i,3)-x3A)*(x(i,3)-x3A);
    //    SyA=SyA+(yx(i,1)-yA)*(yx(i,1)-yA);
end;
Sxx(1,1)=Sx1x1; Sxx(1,2)=Sx1x2;

```



```

//Sxx(1,3)=Sx1x3;
Sxx(2,1)=Sx1x2; Sxx(2,2)=Sx2x2;
//Sxx(2,3)=Sx2x3;
//Sxx(3,1)=Sx1x3; Sxx(3,2)=Sx2x3; Sxx(3,3)=Sx3x3;

Sxx_1=1/Sxx;

z1A=0; z2A=0;
//z3A=0;
for i=1:n
    z1A=z1A+log10(x(i,1)); z2A=z2A+log10(x(i,2));
    //    z3A=z3A+log10(x(i,3));
end;
z1A=z1A/n; z2A=z2A/n;
//z3A=z3A/n;
Sz1z1=0; Sz2z2=0;
//Sz3z3=0;
Sz1z2=0;
//Sz1z3=0; Sz2z3=0;
for i=1:n
    Sz1z1=Sz1z1+(log10(x(i,1))-z1A)*(log10(x(i,1))-z1A);
    Sz1z2=Sz1z2+(log10(x(i,1))-z1A)*(log10(x(i,2))-z2A);
    //    Sz1z3=Sz1z3+(log10(x(i,1))-z1A)*(log10(x(i,3))-z3A);
    //    Sz2z3=Sz2z3+(log10(x(i,2))-z2A)*(log10(x(i,3))-z3A);
    Sz2z2=Sz2z2+(log10(x(i,2))-z2A)*(log10(x(i,2))-z2A);
    //    Sz3z3=Sz3z3+(log10(x(i,3))-z3A)*(log10(x(i,3))-z3A);
end;
Szz(1,1)=Sz1z1/n; Szz(1,2)=Sz1z2/n;
//Szz(1,3)=Sz1z3;
Szz(2,1)=Sz1z2/n; Szz(2,2)=Sz2z2/n;
//Szz(2,3)=Sz2z3;
//Szz(3,1)=Sz1z3; Szz(3,2)=Sz2z3; Szz(3,3)=Sz3z3;

Szz_1=1/Szz;

zi(1)=log10(x1i)-z1A; zi(2)=log10(x2i)-z2A;
//zi(3)=log10(x3i)-z3A;
ziT=zi';
ziSzz_1=ziT*Szz_1;
result=ziSzz_1*zi;

Dfn=2;
Dfd=n-2;
//    Q=0.005;
//    P=1-Q;
Q=1-Pconf;
P=Pconf;
[F]=cdf('F',Dfn,Dfd,P,Q);
test_st=2*(n*n-1)*F/n/(n-2);

```

```

set(textresult,'string', sprintf("=%1.4f",result));
set(textresult2,'string', sprintf("=%1.4f",test_st));
//set(textresult3,'string', sprintf("=%1.4f",yUBC));
//set(textresult4,'string',..
// sprintf("Коефіцієнт детермінації  $R^2$ =%1.4f",r_2));
//set(textresult5,'string', sprintf("=%1.4f",yLBPr));
//set(textresult6,'string', sprintf("=%1.4f",yUBPr));

if result > test_st then
    set(textresult4,'string',sprintf("Складність висока
SMD=%1.4f",result));
else
    set(textresult4,'string',sprintf("Складність не висока
SMD=%1.4f",result));
end;

if (get(radius_bikv_x,'value')==0)

else

end;

endfunction

//Функция закрытия окна.
function W Close()
close(f);
close(f1);
//close(f2);
//close(f3);
endfunction

```

## **ДОДАТОК В – Опис програми**

### **Загальні відомості**

Програма для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java для системи моделювання Scilab 6.0.0. Файл програми Pred\_El\_lg\_CBO\_RFC.sce має розмір 10 КБ.

### **Функціональне призначення**

Програма Pred\_El\_lg\_CBO\_RFC призначений для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java для системи моделювання Scilab 6.0.0.

### **Технічні засоби, що використовуються**

Програма має експлуатуватись на обладнанні з параметрами не нижчими за наступні: процесор Pentium IV (або еквівалентний) з інструкціями SSE2, оперативна пам'ять 2 Гб (1 Гб мінімум), простір на жорсткому диску 600 Мб мінімум.

### **Виклик та завантаження**

Для запуску та роботи з програмою необхідна операційна система Windows Vista/7/8/10 (32 або 64 біти), система моделювання Scilab 6.0.0.

Завантаження програми здійснюється шляхом відкриття файлу Pred\_El\_lg\_CBO\_RFC.sce стандартними засобами ОС.

## ДОДАТОК Г – Інструкція користувача

### Загальний опис роботи з програмою

Після завантаження файлу Pred\_El\_lg\_CBO\_RFC.sce на екрані з'являється форма для відображення початкового стану програми для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java (рисунок Г.1).

Оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java

File Tools Edit ?

Оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java

Дані для оцінювання

Значення СВО на рівні застосунку 4.0

Значення RFC на рівні застосунку 7.5

Довірча ймовірність 0.995

Файл з даними

D:\Data\_F\CBO\_RFC\_44.txt

Ввести

Наявність нормалізації

☒ є (включити), немає (виключити)

Оцінювання

Результати оцінювання

Квадрат відстані Махаланобіса

Тестова статистика на основі F розподілу

Закрити

Рисунок Г.1 – Форма для відображення початкового стану програми

Далі у вікні редагування «Файл з даними» графічного вікна «Оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на

Java» (рис.Г.1) потрібно відредагувати ім'я текстового файлу з початковими емпіричними даними, що будуть використовуватися для оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java.

Після натискання на командну кнопку «Ввести» з текстового файлу вводяться дані.

Потім у графічному вікні «Оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java» (рис. Г.1) у відповідних вікнах редагування потрібно відредагувати: значення метрик CBO та RFC, довірчу ймовірність.

Крім того у графічному вікні «Оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java» (рис. Г.1) потрібно за допомогою радіо кнопки здійснити вибір наявності нормалізації.

Далі потрібно натиснути на командну кнопку «Оцінювання». Після чого з'являться результати оцінювання складності ООП через зв'язки між класами застосунку з відкритим кодом на Java.

І на останок у графічному вікні «Оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java» головного модуля програми (рис. Г.1) потрібно натиснути на командну кнопку «Закрити». Це приведе до закриття всіх графічних вікон та завершення програми.

## **ДОДАТОК Д – Програма та методика випробувань**

**Об’єкт випробувань:** Програма для оцінювання складності ООП через зв’язки між класами застосунків з відкритим кодом на Java Pred\_El\_lg\_CBO\_RFC.

**Ціль випробувань:** перевірка відповідності програми Pred\_El\_lg\_CBO\_RFC до вимог технічного завдання і повноти реалізації.

### **Склад програмної документації:**

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

### **Технічні вимоги.**

Програма має експлуатуватись на обладнанні з параметрами не нижчими за наступні: процесор Pentium IV (або еквівалентний) з інструкціями SSE2, оперативна пам’ять 2 Гб (1 Гб мінімум), простір на жорсткому диску 600 Мб мінімум.

### **Методи випробувань.**

Випробування програми виконується представником замовника в присутності представника розробника. Відповідно до опису програми виконується запуск програми Pred\_El\_lg\_CBO\_RFC. Далі виконується перевірка роботи програми. Виконується перевірка всіх екранних форм програми, а також всіх допустимих для виконання операцій всередині екранних форм згідно до посібника користувача. Якщо програма демонструє адекватне

функціонування згідно до вимог ТЗ та представленої програмної документації, то робота з розробки програми вважається виконаною, і виконується впровадження програми до експлуатації. В протилежному випадку програма має бути відправлено на доробку розробникові.

Використовуються тести за методологією „чорної скриньки” (рис.Д.1-Д.5).

Оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java

File Tools Edit ?

Оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java

Дані для оцінювання

Значення СВО на рівні застосунку 4.0

Значення RFC на рівні застосунку 7.5

Довірча ймовірність 0.995

Файл з даними

D:\Data\_F\CBO\_RFC\_44.txt

Ввести

Наявність нормалізації

☒ є (включити), немає (виключити)

Оцінювання

Результати оцінювання

Квадрат відстані Махаланобіса

Тестова статистика на основі F розподілу

Закрити

Рисунок Д.1 – Графічне вікно головного модуля програми

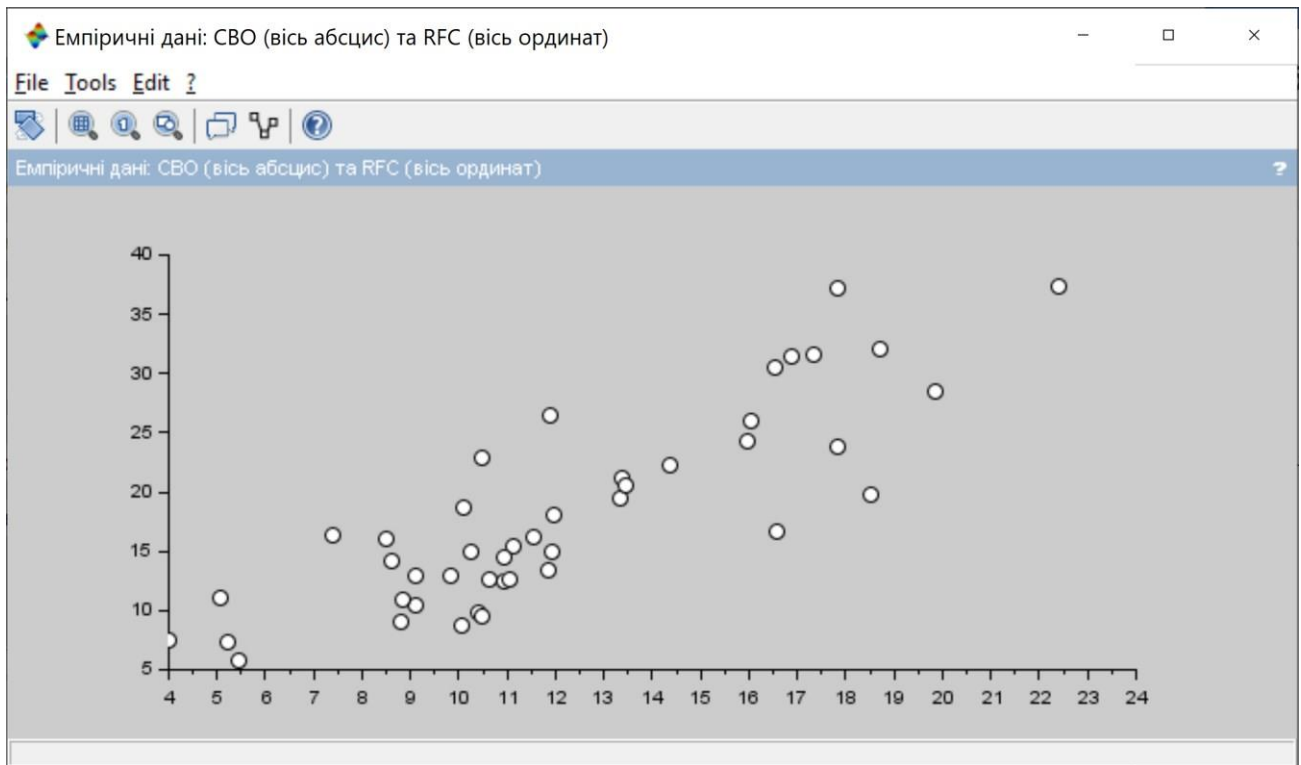


Рисунок Д.2 – Візуалізація тестових емпіричних даних.

Результати роботи програми після натискання на командну кнопку «Оцінювання» у разі, коли радіо кнопка включена та введені значення метрик CBO та RFC застосунку з не високою складністю ООП, наведені на рис. Д.3.



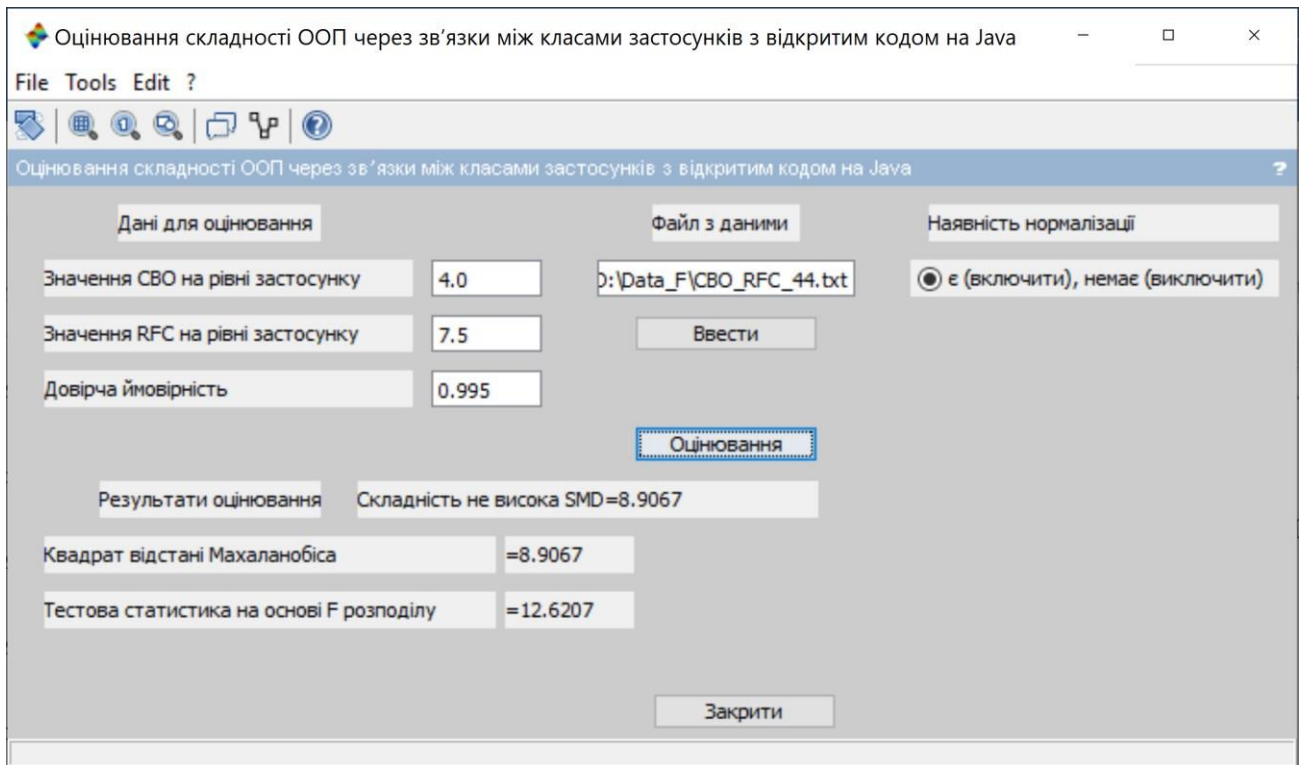


Рисунок Д.3 – Результати роботи програми після натискання на командну кнопку «Оцінювання» у разі, коли радіо кнопка включена та введені значення метрик СВО та RFC застосунку з не високою складністю ООП

Результати роботи програми після натискання на командну кнопку «Оцінювання» у разі, коли радіо кнопка включена та введені значення метрик СВО та RFC застосунку з високою складністю ООП, наведені на рис. Д.4.

Оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java

File Tools Edit ?

Оцінювання складності ООП через зв'язки між класами застосунків з відкритим кодом на Java

Дані для оцінювання

Значення СВО на рівні застосунку: 9.93

Значення RFC на рівні застосунку: 43.23

Довірча ймовірність: 0.995

Файл з даними: D:\Data\_F\CBO\_RFC\_44.txt

Ввести

Наявність нормалізації: ☒ є (включити), немає (виключити)

Оцінювання

Результати оцінювання

Складність висока SMD=19.7108

Квадрат відстані Махаланобіса: =19.7108

Тестова статистика на основі F розподілу: =12.6207

Закрити

Рисунок Д.4 – Результати роботи програми після натискання на командну кнопку «Оцінювання» у разі, коли радіо кнопка включена та введені значення метрик СВО та RFC застосунку з високою складністю ООП

При випробуванні програми потрібно ввести наступні тестові дані із текстового файлу:

```

10.4091 9.8636
11.9038 26.4477
10.047 8.6879
11.1295 15.3894
17.8436 23.8848
9.1124 10.4876
17.3327 31.5262

```

5.0714 11.1429  
22.4167 37.2778  
5.2047 7.3064  
13.3763 21.1303  
18.7195 32.0096  
10.9242 14.5556  
11.8396 13.4284  
8.8377 10.8677  
10.9353 12.4876  
4 7.5  
13.3333 19.4667  
10.4737 22.9474  
8.8077 9.0769  
7.3918 16.3509  
14.3478 22.2145  
9.832 12.9688  
10.2448 15.0196  
18.5265 19.7506  
8.625 14.125  
15.9626 24.2139  
11.9123 14.9219  
11.0559 12.6526  
11.9701 18.1239  
10.4902 9.521  
16.0278 25.9306  
11.5291 16.2646  
10.0773 18.6289  
17.8138 37.1579  
9.1031 12.9736  
10.6263 12.5962  
5.4372 5.7806  
16.535 30.5372  
16.5904 16.6383

16.8825 31.3881  
13.4609 20.5251  
19.8384 28.5483  
8.4792 16.0685