

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

_____ проф. Приходько С.Б.

«__» _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти «бакалавр»

на тему: Розробка програмного забезпечення для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel

Виконав: студент групи 4151

Грабовський Є.В.

(підпис, ПІБ)

Керівник роботи:

Зав.кафедри ПЗАС, д.т.н., професор

(посада, науковий ступень вчене звання)

Приходько С.Б.

(підпис, ПІБ)

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

_____ доц. Макарова Л.М.

(підпис)

«__» _____ 2022 р.

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

Студенту Грабовському Євгену Володимировичу _____

(Прізвище, ім'я, по батькові)

1. Тема роботи: Розробка програмного забезпечення для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel _____

Керівник роботи Приходько С.Б. _____

Затверджені наказом ректора №256-уч від «11» травня 2022 р.

2. Термін подання роботи: 10.06.2022 р. _____

3. Вихідні дані по роботі: _____

4. Перелік питань, що належать до розробки (найменування розділів) _____

Титульний аркуш; Завдання на кваліфікаційну роботу; Анотація (українською та англійською мовами); Зміст; Перелік умовних позначень, символів, одиниць та термінів (при необхідності); Вступ; Опис предметної галузі та постановка задачі; Проект програмного забезпечення (ескізний, технічний та робочий); Результати розробки програмного забезпечення; Розділ з охорони праці; Висновки; Список використаних джерел; Додатки (технічне завдання, текст програми, опис програми, інструкція користувача, програма та методика випробувань програмного забезпечення).

5. Перелік презентаційних матеріалів: Тема роботи, Актуальність теми, Мета роботи, Регресійна модель для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel, Діаграма варіантів використання ПЗ, Діаграма діяльності ПЗ, Графічне вікно програми із інтерфейсними компонентами, Кількість строк коду (у тисячах строк) 30 Web-застосунків в залежності від загальної кількості класів, Кількість строк коду (у тисячах строк) 30 Web-застосунків в залежності від середньої кількості методів на клас, Кількість строк коду (у тисячах строк) 30 Web-застосунків в залежності від суми середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти, Результати оцінювання значень нелінійної регресії, нижньої і верхньої границь довірчого інтервалу нелінійної регресії розміру Web-застосунків, Висновки _____

АНОТАЦІЯ

Кваліфікаційна робота присвячена розробці програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.

Кваліфікаційна робота викладена на 88 сторінках машинописного тексту, містить 15 рисунків, 6 таблиць, 5 додатків та список використаної літератури з 32 найменувань.

ABSTRACT

The qualification paper is devoted to the development of software for estimating the number of lines of code of Web-applications created using the Laravel framework.

The qualification paper is presented on 88 pages of typewritten text, contains 15 figures, 6 tables, 5 appendices and 32 references.

ЗМІСТ

ВСТУП	7
1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ДЛЯ ОЦІНЮВАННЯ КІЛЬКОСТІ СТРОК КОДУ WEB-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ ЗА ДОПОМОГОЮ ФРЕЙМВОРКУ LARAVEL, ТА ПОСТАНОВКА ЗАДАЧІ .	9
1.1 Існуючі методи для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel	9
Засоби реалізації програми для оцінювання кількості строк коду Web- застосунків, що створюються за допомогою фреймворку Laravel	26
1.3 Постановка задачі	28
2 РОЗРОБКА ПРОЕКТУ ПРОГРАМИ ДЛЯ ОЦІНЮВАННЯ КІЛЬКОСТІ СТРОК КОДУ WEB-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ ЗА ДОПОМОГОЮ ФРЕЙМВОРКУ LARAVEL	30
2.1 Розробка ескізного проекту програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel	30
2.2 Розробка технічного проекту програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel	36
2.3 Розробка робочого проекту програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel	40
3 РЕЗУЛЬТАТИ РОЗРОБКИ	48
4 ОХОРОНА ПРАЦІ	56
4.1 Вступ	56
4.2 Аналіз шкідливих та небезпечних факторів при роботі з персональними комп'ютерами	57
4.3 Розрахунок пожежної безпеки на робочому місці користувача персонального комп'ютера	59
4.4 Вимоги безпеки при виконанні робіт	60

ВИСНОВКИ	63
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	64
Додаток А – Технічне завдання на розробку програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel	69
Додаток Б – Текст програми	73
Додаток В – Опис програми	80
Додаток Г – Інструкція користувача	81
Додаток Д – Програма та методика випробувань	83

ВСТУП

Задача оцінювання розміру Web-застосунків як і іншого програмного забезпечення (ПЗ) на ранній стадії розробки є важливою, оскільки ця інформація використовується для прогнозування трудомісткості створення ПЗ за допомогою такої відомої моделі як COCOMO II [1]. У свою чергу зараз для створення Web-застосунків широко використовують фреймворки, серед яких широкою популярністю користується Laravel (<https://laravel.com/>), – безкоштовний PHP фреймворк з відкритим вихідним кодом, створений Тейлором Отвеллом (Taylor Otwell) для розробки Web-застосунків за проектним шаблоном MVC (Model-View-Controller), що зазвичай застосовується для розробки користувацького інтерфейсу, який розділяє відповідну логіку програми на три взаємопов'язані елементи. Зазначимо, що PHP – це популярна мова сценаріїв загального призначення з відкритим вихідним кодом, яка особливо добре підходить для розробки веб-застосунків на стороні сервера. В наш час PHP використовується більш ніж у 80% всіх веб-сайтів, наприклад, таких як Tesla, Wikipedia, WordPress.com [1]. Основне призначення цієї мови полягає у тому, щоб дозволити веб-розробникам швидко писати веб-сторінки, що генеруються динамічно. Зазначене вище вказує на те, що задача оцінювання розміру Web-застосунків, що створюються за допомогою фреймворку Laravel, є практичною задачею інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов.

Крім того, існуюче ПЗ, яке застосовується для оцінювання залежної змінної регресії (наприклад, MatLab, Scilab), не має у своєму складі потрібних функцій, які дозволяли виконувати відповідне оцінювання. Тому виникає потреба у створенні оригінальної програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.

Метою кваліфікаційної роботи є розробка проекту саме такої програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.

Для досягнення цієї мети необхідно вирішити наступні задачі.

- 1) Виконати аналіз існуючих методів для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.
- 2) Зробити постановку задачі на розробку програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.
- 3) Розробити ескізний проект програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.
- 4) Розробити технічний проект програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.
- 5) Розробити робочий проект програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.

1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ДЛЯ ОЦІНЮВАННЯ КІЛЬКОСТІ СТРОК КОДУ WEB-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ ЗА ДОПОМОГОЮ ФРЕЙМВОРКУ LARAVEL, ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Існуючі методи для оцінювання кількості строк коду Web- застосунків, що створюються за допомогою фреймворку Laravel

При оцінюванні трудомісткості розробки різноманітного ПЗ, в тому числі і Web-застосунків, часто використовуються методи нелінійного регресійного аналізу, нелінійні регресійні моделі та рівняння, зокрема COCOMO II [2-5], основним фактором в яких є розмір ПЗ. Але, як правило, розмір програми на початковому етапі її розробки не відомий, що потребує його оцінювання. В [6, 7] для оцінювання кількості строк коду інформаційних РНР-систем з відкритим кодом Hee Beng Kuan Tan, Yuan Zhao та Hongyu Zhang запропонували лінійне регресійне рівняння в залежності від трьох метрик концептуальної моделі даних у вигляді діаграми класів. Це рівняння було побудоване на основі методів множинного лінійного регресійного аналізу. Але, як відомо, при побудові лінійних регресійних моделей необхідно виконання певних умов, зокрема похибки повинні бути розподілені за нормальним законом, що має місце лише в поодиноких випадках. А це веде до необхідності побудови нелінійних регресійних моделей для оцінювання кількості строк ПЗ та застосування відповідних методів множинного нелінійного регресійного аналізу.

Нормалізуючі перетворення часто є добрим способом побудови моделей, рівнянь, довірчих інтервалів та інтервалів прогнозування нелінійних регресій [8-18]. В роботах [16-21] запропоновано рівняння нелінійної регресії для оцінювання розміру ПЗ інформаційних системи на РНР. Запропоноване рівняння побудовано за допомогою множинного нелінійного регресійного

аналізу із застосуванням чотиривимірного перетворення Джонсона сім'ї S_B на основі таких же метрик діаграми класів, що і в [6, 7]: загальна кількість класів, загальна кількість зв'язків та середня кількість атрибутів на клас. Але для Web-застосунків, що не є інформаційними системами, наприклад, таких як різноманітні фреймворки, конвертори, HTTP-клієнти та інші, регресійні моделі можуть залежати в тому числі від інших метрик.

Зазвичай при побудові нелінійних регресійних моделей у якості нормалізуючого перетворення використовують десятковий логарифм. Наприклад, такі відомі моделі як COCOMO, ISBSG, COCOMO II були створені саме із використанням нормалізуючого перетворення у вигляді десятичного логарифму. Тому в [22] було розроблено нелінійну регресійну модель для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel саме на основі нормалізуючого перетворення у вигляді десятичного логарифму Цю модель ми і будемо використовувати.

Існуюче ПЗ, яке застосовується для оцінювання залежної змінної нелінійної регресії (наприклад, MatLab, Scilab), не має у своєму складі потрібних функцій, які дозволяли виконувати відповідне оцінювання. Тому виникає потреба у створенні оригінальної програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.

Якщо випадкові величини Z_1 і Z_Y мають нормальний розподіл, то проблем з побудовою рівняння регресії немає тому, що воно буде лінійним

$$\hat{Z}_Y = b_0 + b_1 Z_1. \quad (1.1)$$

В загальному випадку k незалежних змінних (регресорів) лінійна регресійна модель є такою:

$$Z_Y = \hat{Z}_Y + \varepsilon = \bar{Z}_Y + (\mathbf{Z}_X^+) \hat{\mathbf{b}} + \varepsilon, \quad (1.2)$$

де $\hat{Z}_Y$ – результат передбачення рівняння лінійної регресії для значень компонент вектору $\mathbf{z}_X = \{Z_1, Z_2, \dots, Z_k\}$; ε – випадкова величина з розподілом Гаусу, $\varepsilon \sim N(0, \sigma_\varepsilon^2)$; $\mathbf{Z}_X^+$ – матриця центрованих регресорів, яка містить

значення $Z_{1_i} - \bar{Z}_1, Z_{2_i} - \bar{Z}_2, \dots, Z_{k_i} - \bar{Z}_k$; $\hat{\mathbf{b}}$ – оцінка для вектору параметрів рівняння лінійної регресії, $\mathbf{b} = \{b_1, b_2, \dots, b_k\}^T$.

У разі нормального розподілу не виникає проблем і з побудовою довірчого інтервалу лінійної регресії. Цей інтервал визначається із використанням квантилів розподілу Стюдента [11]

$$\hat{Z}_{Y_i} \pm t_{\alpha/2, N-2} S_{Z_Y} \sqrt{\frac{1}{N} + \frac{(Z_{1_i} - \bar{Z}_1)^2}{S_{Z_1 Z_1}}},$$

де $\hat{Z}_{Y_i}$ – значення $\hat{Z}_Y$ за рівнянням (1.1) за значенням Z_{1_i} ; Z_{1_i} – i -те значення Z_1 ; $t_{\alpha/2, N-2}$ – квантиль t -розподілу Стюдента з $N - 2$ степенями вільності та

$\alpha/2$ рівнем значимості; $S_{Z_1 Z_1} = \sum_{i=1}^N (Z_{1_i} - \bar{Z}_1)^2$; $\bar{Z}_1 = \frac{1}{N} \sum_{i=1}^N Z_{1_i}$;

$$S_{Z_Y}^2 = \frac{1}{N-1} \sum_{i=1}^N (Z_{Y_i} - \bar{Z}_Y)^2; \bar{Z}_Y = \frac{1}{N} \sum_{i=1}^N Z_{Y_i}.$$

Для випадку k незалежних змінних довірчий інтервал лінійної регресії може бути записаний таким чином [11]:

$$\hat{Z}_Y \pm t_{\alpha/2, v} S_{Z_Y} \left\{ \frac{1}{N} + (\mathbf{z}_X^+)^T \left[(\mathbf{z}_X^+)^T \mathbf{z}_X^+ \right]^{-1} (\mathbf{z}_X^+) \right\}^{1/2}, \quad (1.3)$$

а інтервал передбачення лінійної регресії як

$$\hat{Z}_Y \pm t_{\alpha/2, v} S_{Z_Y} \left\{ 1 + \frac{1}{N} + (\mathbf{z}_X^+)^T \left[(\mathbf{z}_X^+)^T \mathbf{z}_X^+ \right]^{-1} (\mathbf{z}_X^+) \right\}^{1/2}, \quad (1.4)$$

де $S_{Z_Y}^2 = \frac{1}{v} \sum_{i=1}^N (Z_{Y_i} - \hat{Z}_{Y_i})^2$, $v = N - k - 1$; $(\mathbf{z}_X^+)^T \mathbf{z}_X^+ - k \times k$ матриця

$$(\mathbf{z}_X^+)^T \mathbf{z}_X^+ = \begin{pmatrix} S_{Z_1 Z_1} & S_{Z_1 Z_2} & \dots & S_{Z_1 Z_k} \\ S_{Z_1 Z_2} & S_{Z_2 Z_2} & \dots & S_{Z_2 Z_k} \\ \dots & \dots & \dots & \dots \\ S_{Z_1 Z_k} & S_{Z_2 Z_k} & \dots & S_{Z_k Z_k} \end{pmatrix},$$

де $S_{Z_q Z_r} = \sum_{i=1}^N [Z_{q_i} - \bar{Z}_q][Z_{r_i} - \bar{Z}_r]$, $q, r = 1, 2, \dots, k$.

У разі, якщо випадкові величини будуть мати негаусівський розподіл, рівняння та модель регресії будуть нелінійними. Але яку саме не лінійність потрібно застосовувати заздалегідь, як правило, не відомо. Тому на сьогодні для побудови нелінійного рівняння регресії та нелінійної регресійної моделі в основному використовують метод простого перебору. За цим методом беруть різні не лінійності, будують сукупність нелінійних моделей регресії і з цієї сукупності за певним критерієм обирають найкраще рівняння чи найкращу модель. Але застосування методу простого перебору потребує багато часу і не гарантує знаходження найкращого рівняння нелінійної регресії чи нелінійної регресійної моделі тому, що вони в цю сукупність можуть не потрапити. Із-за цього для побудови нелінійного рівняння регресії чи нелінійної регресійної моделі прагнуть використовувати інші методи, які не потребують простого перебору. До цих методів відносять методи, що базуються на лінеаризуючих та нормалізуючих перетвореннях. Методи, що використовують лінеаризацію, можуть застосовуватися лише в поодиноких випадках, коли вдається знайти лінеаризуюче перетворення. І навпаки, нормалізуюче перетворення, як правило, завжди вдається підібрати, що дає можливість застосовувати для побудови нелінійної регресійної моделі саме метод на основі нормалізуючих перетворень. Суть цього методу полягає у наступному. Спочатку потрібно нормалізувати емпіричні дані, для чого треба здійснити вибір нормалізуючого перетворення і оцінювання його параметрів. Далі за нормалізованими даними отримати лінійну регресійну модель. І на останок, використовуючи обране нормалізуюче перетворення та побудоване лінійне рівняння регресії для нормалізованих даних, здійснити перехід до нелінійної регресійної моделі [8].

Отже за [8] нелінійна регресійна модель може бути побудована за допомогою нормалізуючого перетворення і записана як

$$Y = \psi_Y^{-1} [\bar{Z}_Y + (\mathbf{Z}_X^+) \hat{\mathbf{b}} + \varepsilon], \quad (1.5)$$

де ψ_Y – перша компонента вектору Ψ , $\Psi = \{\psi_Y, \psi_1, \psi_2, \dots, \psi_k\}^T$,

нормалізуючого перетворення

$$\mathbf{T} = \boldsymbol{\psi}(\mathbf{P}), \quad (1.6)$$

яке має зворотнє перетворення

$$\mathbf{P} = \boldsymbol{\psi}^{-1}(\mathbf{T}). \quad (1.7)$$

В (1.6) та (1.7) $\mathbf{P} = \{Y, X_1, X_2, \dots, X_k\}^T$ – це негаусівський вектор, а $\mathbf{T} = \{Z_Y, Z_1, Z_2, \dots, Z_k\}^T$ – гаусівський вектор.

Аналогічно із застосуванням нормалізуючих перетворень в роботі [8] було запропоновано визначати довірчий інтервал нелінійної регресії за (1.3) як

$$\boldsymbol{\psi}_Y^{-1} \left(\hat{Z}_Y \pm t_{\alpha/2, \nu} S_{Z_Y} \left\{ \frac{1}{N} + (\mathbf{z}_X^+)^T [(\mathbf{Z}_X^+)^T \mathbf{Z}_X^+]^{-1} (\mathbf{z}_X^+) \right\}^{1/2} \right), \quad (1.8)$$

а інтервал передбачення нелінійної регресії за (1.4) як

$$\boldsymbol{\psi}_Y^{-1} \left(\hat{Z}_Y \pm t_{\alpha/2, \nu} S_{Z_Y} \left\{ 1 + \frac{1}{N} + (\mathbf{z}_X^+)^T [(\mathbf{Z}_X^+)^T \mathbf{Z}_X^+]^{-1} (\mathbf{z}_X^+) \right\}^{1/2} \right). \quad (1.9)$$

Практичне застосування нормалізуючих перетворень для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel потребує пошуку відповідного перетворення. У якості нормалізуючого перетворення використовується перетворення у вигляді десяткового логарифму.

Для побудови нелінійної регресійної моделі використовувалися такі емпіричні дані [22]: фактичний розмір Web-застосунків в тисячах рядків коду Y , загальна кількість класів X_1 , середня кількість методів на клас X_2 та суму середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти, X_3 . Ці дані наведені у таблиці 1.1.

Таблиця 1.1

Набір даних та значення SMD

№	Y	X_1	X_2	X_3	SMD
1	2	3	4	5	6
1	20,816	171	9,053	2,485	3,76
2	9,482	180	3,100	4,644	3,54
3	57,454	1000	4,039	4,612	24,18

Продовження табл. 1.1

1	2	3	4	5	6
4	19,481	148	9,939	2,466	4,69
5	3,532	75	2,773	3,173	0,50
6	17,888	98	13,847	1,500	15,52
7	3,825	81	2,728	2,926	0,42
8	22,438	308	5,948	4,127	1,13
9	1,881	44	3,000	2,773	0,56
10	2,364	44	3,114	3,091	0,62
11	67,055	727	4,455	4,344	21,63
12	9,305	146	3,466	3,158	0,08
13	3,364	97	2,299	3,124	0,57
14	0,47	23	0,913	1,478	4,78
15	3,391	76	2,250	3,500	1,11
16	4,069	107	1,879	2,607	1,05
17	7,241	194	3,108	3,010	1,71
18	0,515	21	1,238	1,714	3,56
19	1,771	57	2,175	2,877	0,70
20	16,817	185	11,000	5,119	11,79
21	1,072	40	1,925	2,500	1,14
22	10,774	184	3,065	4,038	1,21
23	17,743	313	3,994	4,550	2,33
24	1,044	42	1,905	2,952	0,93
25	0,805	29	1,552	2,000	2,41
26	8,618	215	3,019	3,563	1,40
27	33,591	365	6,312	4,014	3,13
28	28,361	303	6,535	3,743	1,97
29	5,732	88	3,500	4,068	2,35
30	8,615	210	3,043	3,490	1,23

Метрики (майбутні фактори) X_1 , X_2 та X_3 були перевірені на наявність мультиколінеарності, яку ми визначали за коефіцієнтами впливу дисперсії (VIFs) серед майбутніх факторів в моделі множинної лінійної регресії. VIFs – це діагональні елементи оберненої коваріаційної матриці k -факторів. Значення VIFs більше за 10 часто сприймаються як сигнал, що дані мають проблеми з

мультиколінеарністю. У разі, якщо значення VIFs знаходяться у межах від 1 до 5, то мультиколінеарності немає. Значення VIFs для X_1 , X_2 , та X_3 дорівнюють 1,56, 1,03 та 1,52 відповідно, що вказує на відсутність мультиколінеарності.

Відповідно з [22] розподіл даних з таблиці 1.1 є негаусівським тому, що для трьох застосунків (3, 6 та 11) значення SMD, які відповідно дорівнюють 24,18, 15,52 та 21,63, є більшими ніж величина квантіля розподілу χ^2 , що становить 14,86 для рівня значущості 0,005 та чотирьох ступенів свободи. Про негаусівський розподіл даних з табл. 1 також свідчить оцінка багатовимірного ексцесу, що дорівнює 52,09. Це значення більш ніж вдвічі перевищує теоретичне, що у нашому випадку дорівнює 24 згідно з [23].

Враховуючи негаусівський розподіл чотиривимірних даних з таблиці 1.1, в подальшому для виявлення викидів у цих даних з таблиці 1.1 ми використовуємо метод, оснований на нормалізуючих перетвореннях і квадраті відстані Махаланобіса SMD [25, 26], який для кожної багатовимірної точки нормалізованих даних i , $i = 1, 2, \dots, N$, визначається як

$$d_i^2 = (\mathbf{T}_i - \bar{\mathbf{T}})^T \mathbf{S}_N^{-1} (\mathbf{T}_i - \bar{\mathbf{T}}), \quad (1.10)$$

де $\bar{\mathbf{T}}$ – вектор вибірових середніх нормалізованих даних; $\mathbf{S}_N$ – вибіркова коваріаційна матриця нормалізованих даних

$$\mathbf{S}_N = \frac{1}{N} \sum_{i=1}^N (\mathbf{T}_i - \bar{\mathbf{T}})(\mathbf{T}_i - \bar{\mathbf{T}})^T. \quad (1.11)$$

Тут $\mathbf{T}_i$ – гаусівський випадковий вектор $\mathbf{T} = \{Z_Y, Z_1, Z_2, \dots, Z_k\}^T$ для кожної багатовимірної точки нормалізованих даних i , $i = 1, 2, \dots, N$. В нашому випадку кількість факторів k дорівнює 3. Вектор $\mathbf{T}$ отримують шляхом перетворення негаусівського випадкового вектору $\mathbf{P} = \{Y, X_1, X_2, \dots, X_k\}^T$ за допомогою багатовимірного нормалізуючого перетворення

$$\mathbf{T} = \boldsymbol{\psi}(\mathbf{P}). \quad (1.12)$$

В нашому випадку, що розглядається, в перетворенні (1.12) $\boldsymbol{\psi} = \{\lg Y, \lg X_1, \lg X_2, \lg X_3\}^T$.

Зазначимо, що перетворення (1.12) є взаємо-зворотнім, тобто для (1.12) існує зворотне перетворення

$$\mathbf{P} = \Psi^{-1}(\mathbf{T}). \quad (1.13)$$

Відомо [13], якщо дані мають багатовимірний нормальний розподіл, то розподіл SMD поводить ся як розподіл χ^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 χ^2 з рівнем значимості α . Значення даних, для яких величина SMD більша, ніж квантиль розподілу χ^2 , розглядаються як викиди, і ці значення відкидаються. Після відкидання зменшене число багатомірних точок даних знову нормалізується, використовуючи перетворення (1.12), доки всі значення SMD (1.10) будуть меншими чи рівними квантилю розподілу χ^2 .

Для рівня значимості 0,005 та перетворення у вигляді десятичного логарифму викиди відсутні тому, що для всіх рядків даних SMD є меншим ніж величина квантіля розподілу χ^2 , яка становить 14,86 для рівня значимості 0,005.

В подальшому чотиривимірні негаусові дані, що наведені у таблиці 1.1, були використані для побудови нелінійної регресійної моделі для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.

Але спочатку для остаточного обґрунтування необхідності її побудови отримуємо лінійну регресійну модель для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel, яка представляється у вигляді

$$\hat{Y} = \hat{b}_0 + \hat{b}_1 X_1 + \hat{b}_2 X_2 + \hat{b}_3 X_3 + \varepsilon, \quad (1.14)$$

де оцінки параметрів такі: $\hat{b}_0 = -1,73631$, $\hat{b}_1 = 0,07090$, $\hat{b}_2 = 1,27302$, $\hat{b}_3 = -1,15451$; ε – випадкова величина з розподілом Гаусу, $\varepsilon \sim N(0, \sigma_\varepsilon^2)$ з оцінкою дисперсії $\hat{\sigma}_\varepsilon^2 = 19,74$.

Перевіримо нульову гіпотезу про нормальність закону розподілу випадкової величини ε для моделі (1.14). Для цього будемо використовувати критерій Пірсона.

У разі великої вибірки значень випадкової величини (коли $n > 30$) перевірку нормальності закону розподілу результатів спостережень в тому числі виконують за критерієм Пірсона (критерієм χ^2). Відповідно за цим критерієм спочатку обчислюють значення χ^2 [27]

$$\chi^2 = \sum_{j=1}^m \frac{(n_j - np_j)^2}{np_j}, \quad (1.15)$$

де m – кількість підінтервалів (часток, кліток), на які розбивається інтервал $[x_{\min}, x_{\max}]$; $x_{\min}$ і $x_{\max}$ – відповідно мінімальне і максимальне значення випадкової величини X ; n_j – абсолютна частота в j -му підінтервалі (кількість значень випадкової величини, які попадають у j -ий підінтервал); p_j – ймовірність того, що значення випадкової величини X попадають у j -ий підінтервал.

Кількість підінтервалів (кліток), на які розбивається інтервал $[x_{\min}, x_{\max}]$ може бути визначений за наступними формулами: $m = \log_2 n + 1 = 3,3 \lg n + 1$ (формула Старджеса) або $m = 5 \lg n$ (формула Брукса і Карузера). Ми використовуємо формулу Старджеса, за якою $m = 6$.

Ймовірність того, що значення випадкової величини X попадають у j -ий підінтервал можуть бути визначені як

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

де x_{j-1} і x_j – ліва і права границі j -го підінтервалу; $f(x)$ – функція щільності ймовірності.

Функція щільності ймовірності нормального розподілу (розподілу Гауса) є такою:

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

де m_x – математичне сподівання випадкової величини X .

Після цього в залежності від рівня значимості α та кількості ступенів вільності ν за таблицею верхніх 100α %-вих точок розподілу χ^2 (знаходять значення $\chi_{кр}^2$). Кількість ступенів вільності ν визначається як

$$\nu = m - k - 1,$$

де k – кількість параметрів, від яких залежить закон розподілу. Для нормального розподілу $k = 2$.

Якщо $\chi^2 \leq \chi_{кр}^2$, то з ймовірністю $1 - \alpha$ можна прийняти гіпотезу про те, що закон розподілу результатів спостережень є нормальним, якщо $\chi^2 > \chi_{кр}^2$ – цю гіпотезу потрібно відкинути.

Для вибірки значень випадкової величини ε значення χ^2 дорівнює 52,04 за формулою (1.15), а $\chi_{кр}^2$ для 3 ступенів вільності та 0,05 рівня значимості дорівнює 7,81. Тобто $\chi^2 > \chi_{кр}^2$, і цю гіпотезу про нормальність розподілу випадкової величини ε потрібно відкинути.

У свою чергу це свідчить про відсутність теоретичного обґрунтування використання моделі лінійної регресії для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel. Це призводить до необхідності побудови нелінійної регресійної моделі для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel за даними, що наведені у таблиці 1.1.

Для побудови нелінійної регресійної моделі для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel спочатку чотиривимірні негаусівські дані з таблиці 1.1 ми нормалізуємо за допомогою одновимірного перетворення у вигляді десятичного логарифму. Далі для нормалізованих даних будемо лінійну

регресійну модель

$$Z_Y = \hat{Z}_Y + \varepsilon = \hat{b}_0 + \hat{b}_1 Z_1 + \hat{b}_2 Z_2 + \hat{b}_3 Z_3 + \varepsilon, \quad (1.16)$$

де $Z_Y = \lg Y$; $Z_j = \lg X_j$, $j = 1, 2, 3$.

Параметри моделі (1.16) оцінювалися методом найменших квадратів. Оцінки параметрів для (1.16) такі: $\hat{b}_0 = -1,74968$, $\hat{b}_1 = 1,07738$, $\hat{b}_2 = 0,76106$, $\hat{b}_3 = -0,19497$.

На останок будуємо нелінійну регресійну модель за лінійним рівнянням (1.16) та перетворенням у вигляді десяткового логарифму

$$Y = 10^{\varepsilon + \hat{b}_0} X_1^{\hat{b}_1} X_2^{\hat{b}_2} X_3^{\hat{b}_3}. \quad (1.17)$$

Рівняння (1.17) – це і є нелінійна регресійна модель для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.

Побудована модель (1.17) була перевірена за множинним коефіцієнтом детермінації R^2 , середньою величиною відносної помилки MMRE і відсотком прогнозованих результатів, для яких величини відносної помилки MRE менші за 0,25, PRED(0,25). Ці показники зазвичай використовуються для оцінювання якості прогнозування за допомогою регресійних моделей і в інженерії програмного забезпечення [28, 29]. Допустимі значення MMRE і PRED(0,25) складають не більше 0,25 і не менше 0,75 відповідно. Допустиме значення множинного коефіцієнта детермінації R^2 приблизно таке ж, як для PRED(0,25).

Для моделі (1.17) значення R^2 , MMRE і PRED(0,25), які дорівнюють відповідно 0,946, 0,135 і 0,900, є задовільними і кращими за значення цих показників для лінійної моделі (1.14), які дорівнюють відповідно 0,926, 0,382 і 0,600. Сума квадратів відхилень SSR для моделі (1.17) складає 404,9, що на 41 % менше у порівнянні з відповідною сумою для моделі (1.14).

Для визначення нижньої і верхньої границь інтервалів передбачення нелінійних регресій використовуємо відповідні рівняння [8]

$$\hat{Y}_{PI} = \psi_Y^{-1} \left(\hat{Z}_Y \pm t_{\alpha/2, \nu} S_{Z_Y} \left\{ 1 + \frac{1}{N} + (\mathbf{z}_X^+)^T \left[(\mathbf{Z}_X^+)^T \mathbf{Z}_X^+ \right]^{-1} (\mathbf{z}_X^+) \right\}^{1/2} \right). \quad (1.18)$$

В (1.18) знаки мінус та плюс відповідають рівнянням нижньої та верхньої границь інтервалів передбачення нелінійних регресій; $t_{\alpha/2, \nu}$ – квантіль t -розподілу Стюдента з кількістю ступенів вільності ν та рівнем значущості $\alpha/2$; $\mathbf{Z}_X^+$ – матриця центрованих регресорів, яка містить значення $Z_{1i} - \bar{Z}_1$,

$$Z_{2i} - \bar{Z}_2, \quad Z_{3i} - \bar{Z}_3; \quad \mathbf{z}_X^+ = \{Z_{1i} - \bar{Z}_1, Z_{2i} - \bar{Z}_2, Z_{3i} - \bar{Z}_3\}^T; \quad S_{Z_Y}^2 = \frac{1}{\nu} \sum_{i=1}^N (Z_{Yi} - \hat{Z}_{Yi})^2,$$

$\nu = N - k - 1$; $(\mathbf{Z}_X^+)^T \mathbf{Z}_X^+ - k \times k$ матриця

$$(\mathbf{Z}_X^+)^T \mathbf{Z}_X^+ = \begin{pmatrix} S_{Z_1 Z_1} & S_{Z_1 Z_2} & \dots & S_{Z_1 Z_k} \\ S_{Z_1 Z_2} & S_{Z_2 Z_2} & \dots & S_{Z_2 Z_k} \\ \dots & \dots & \dots & \dots \\ S_{Z_1 Z_k} & S_{Z_2 Z_k} & \dots & S_{Z_k Z_k} \end{pmatrix}.$$

де $S_{Z_q Z_r} = \sum_{i=1}^N [Z_{qi} - \bar{Z}_q][Z_{ri} - \bar{Z}_r]$, $q, r = 1, 2, \dots, k$. У нашому випадку $k=3$.

Нижні і верхні границі інтервалів передбачення нелінійних регресій для нелінійної регресійної моделі (1.18) наведені в таблиці 1.2. Для нелінійної регресійної моделі (1.18) з оцінками параметрів, що були отримані за даними з таблиці 1.1 з 30 Web-застосунків виявилось: значення Y для жодного застосунку не виходять за визначені межі інтервалу передбачення, що визначалися за (1.18). В таблиці 1.2 ліва границя інтервалу передбачення, що отриманий на першій ітерації за (1.18), позначена як LB_1 , а права – як UB_1 .

Таблиця 1.2

Нижні і верхні границі інтервалу передбачення нелінійної регресії

№	Y	LB_1	UB_1
1	2	3	4
1	20,816	13,878	29,650
2	9,482	5,807	12,137
3	57,454	43,791	97,155

Продовження табл. 1.2

1	2	3	4
4	19,481	12,728	27,383
5	3,532	2,256	4,637
6	17,888	10,991	26,230
7	3,825	2,465	5,044
8	22,438	17,485	36,210
9	1,881	1,374	2,866
10	2,364	1,376	2,904
11	67,055	34,345	74,066
12	9,305	5,501	11,241
13	3,364	2,591	5,316
14	0,47	0,302	0,674
15	3,391	1,907	3,952
16	4,069	2,534	5,297
17	7,241	6,896	14,280
18	0,515	0,342	0,735
19	1,771	1,420	2,928
20	16,817	14,810	33,409
21	1,072	0,905	1,878
22	10,774	6,099	12,576
23	17,743	12,866	26,745
24	1,044	0,913	1,907
25	0,805	0,563	1,186
26	8,618	7,304	15,069
27	33,591	22,052	45,809
28	28,361	18,804	38,962
29	5,732	3,010	6,346
30	8,615	7,195	14,836

На першій ітерації для нормалізованих за перетворенням у вигляді десятичного логарифму даних з 30 застосунків матриця $(\mathbf{Z}_x^+)^T \mathbf{Z}_x^+$ для визначення нижньої і верхньої границь інтервалів передбачення нелінійної регресії за (1.18) є такою:

$$(\mathbf{Z}_X^+)^T \mathbf{Z}_X^+ = \begin{pmatrix} 5,03702 & 2,01916 & 1,21246 \\ 2,01916 & 2,22142 & 0,35084 \\ 1,21246 & 0,35084 & 0,57166 \end{pmatrix},$$

а зворотна матриця є такою:

$$\left[(\mathbf{Z}_X^+)^T \mathbf{Z}_X^+ \right]^{-1} = \begin{pmatrix} 0,60441 & -0,38428 & -1,04608 \\ -0,38428 & 0,74281 & 0,35916 \\ -1,04608 & 0,35916 & 3,74573 \end{pmatrix}.$$

Також були знайдені довірчі інтервали регресій кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel. Для визначення нижньої і верхньої границь довірчих інтервалів нелінійних регресій використовуємо відповідні рівняння [8]

$$\hat{Y}_{CI} = \psi_Y^{-1} \left(\hat{Z}_Y \pm t_{\alpha/2, v} S_{Z_Y} \left\{ \frac{1}{N} + (\mathbf{z}_X^+)^T \left[(\mathbf{Z}_X^+)^T \mathbf{Z}_X^+ \right]^{-1} (\mathbf{z}_X^+) \right\}^{1/2} \right). \quad (1.19)$$

Оцінки $\hat{Y}$, значення нижніх LB та верхніх UB границь довірчих інтервалів лінійної та нелінійної регресій розміру 30 Web-застосунків наведені в таблиці 1.3. Межі довірчих інтервалів визначалися за (1.3) та (1.19).

Таблиця 1.3

Оцінки $\hat{Y}$, нижні та верхні границі довірчих інтервалів лінійної та нелінійної регресій кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel

No	Y	Лінійна регресійна модель			Нелінійна регресійна модель		
		$\hat{Y}$	LB	UB	$\hat{Y}$	LB	UB
1	2	3	4	5	6	7	8
1	20,816	19,043	15,270	22,815	20,285	17,554	23,440
2	9,482	9,610	5,860	13,361	8,395	7,501	9,395
3	57,454	68,982	61,466	76,497	65,226	54,016	78,764
4	19,481	18,563	14,377	22,748	18,669	16,013	21,765
5	3,532	3,448	1,291	5,605	3,235	2,983	3,507
6	17,888	21,108	14,070	28,146	16,980	13,134	21,951
7	3,825	4,102	2,007	6,197	3,526	3,288	3,782

Продовження табл. 1.3

1	2	3	4	5	6	7	8
8	22,438	22,909	20,345	25,473	25,162	22,849	27,709
9	1,881	2,001	-0,201	4,203	1,985	1,780	2,213
10	2,364	1,779	-0,449	4,006	1,999	1,759	2,271
11	67,055	50,465	45,420	55,511	50,436	43,133	58,975
12	9,305	9,382	7,554	11,210	7,863	7,353	8,409
13	3,364	4,461	2,303	6,620	3,711	3,437	4,007
14	0,47	-0,650	-4,860	3,560	0,451	0,371	0,548
15	3,391	2,476	-0,058	5,009	2,745	2,490	3,027
16	4,069	5,231	2,753	7,709	3,664	3,274	4,101
17	7,241	12,500	10,523	14,477	9,923	9,013	10,926
18	0,515	-0,650	-4,383	3,082	0,501	0,430	0,584
19	1,771	1,753	-0,544	4,049	2,039	1,869	2,225
20	16,817	19,474	13,268	25,679	22,244	18,111	27,320
21	1,072	0,664	-1,908	3,236	1,304	1,180	1,440
22	10,774	10,550	7,936	13,163	8,758	8,022	9,563
23	17,743	20,287	17,239	23,335	18,550	16,730	20,569
24	1,044	0,258	-2,166	2,681	1,320	1,180	1,475
25	0,805	-0,014	-3,248	3,220	0,817	0,722	0,925
26	8,618	13,237	11,246	15,227	10,491	9,597	11,468
27	33,591	27,544	24,925	30,164	31,783	28,697	35,200
28	28,361	23,745	21,346	26,143	27,068	24,555	29,837
29	5,732	4,262	1,095	7,429	4,370	3,852	4,959
30	8,615	12,997	11,051	14,943	10,332	9,463	11,281

Як можна побачити з таблиці 1.3, для 9 рядків даних (9, 10, 14, 15, 18, 19, 21, 24 та 25) нижні границі довірчих інтервалів лінійної регресії кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel, мають від'ємні значення. Всі нижні границі довірчих інтервалів нелінійної регресії кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel, додатні. Більше того для трьох рядків даних (14, 18 і 25) оцінки $\hat{Y}$ лінійної регресії є від'ємними. Всі оцінки $\hat{Y}$ нелінійної регресії кількості строк коду Web-застосунків, що створюються за допомогою

фреймворку Laravel додатні. Крім того ширини довірчих інтервалів нелінійної регресії виявилися меншими у порівнянні з відповідними інтервалами лінійної регресії для 26 Web-застосунків (1, 2, 4-10, 12-26, 29 і 30). Причому найбільша різниця (4764%) виявилася для застосунку з номером 18 з невеликою кількістю строк коду – 815 (0,815 KLOC), а найменша різниця (5,5%) – для застосунку з номером 8 з 22438 строками коду (22,438 KLOC).

Також були знайдені інтервали передбачення регресій розміру 30 Web-застосунків за (1.3) і (1.19). Відповідні значення нижніх LB та верхніх UB границь інтервалів передбачення лінійної та нелінійної регресій кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel, наведені в таблиці 1.4.

Як можна побачити з таблиці 1.4, для 16 рядків даних (2, 5, 7, 9, 10, 12-16, 18, 19, 21, 24, 25 та 29) нижні границі інтервалів передбачення лінійної регресій кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel, мають від’ємні значення. Всі нижні границі інтервалів передбачення нелінійної регресій кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel, додатні.

Таблиця 1.4

Нижні та верхні границі інтервалів передбачення лінійної та нелінійної регресій розміру PHP-застосунків

No	Y	Лінійна регресійна модель		Нелінійна регресійна модель	
		LB	UB	LB	UB
1	2	3	4	5	6
1	20,816	8,686	29,399	13,878	29,650
2	9,482	-0,738	19,959	5,807	12,137
3	57,454	56,755	81,209	43,791	97,155
4	19,481	8,049	29,076	12,728	27,383
5	3,532	-6,435	13,331	2,256	4,637
6	17,888	9,168	33,047	10,991	26,230
7	3,825	-5,768	13,972	2,465	5,044

Продовження табл. 1.4

1	2	3	4	5	6
8	22,438	12,929	32,889	17,485	36,210
9	1,881	-7,892	11,894	1,374	2,866
10	2,364	-8,120	11,677	1,376	2,904
11	67,055	39,580	61,350	34,345	74,066
12	9,305	-0,435	19,198	5,501	11,241
13	3,364	-5,422	14,345	2,591	5,316
14	0,470	-11,173	9,874	0,302	0,674
15	3,391	-7,496	12,448	1,907	3,952
16	4,069	-4,727	15,189	2,534	5,297
17	7,241	2,655	22,345	6,896	14,280
18	0,515	-10,992	9,691	0,342	0,735
19	1,771	-8,162	11,667	1,420	2,928
20	16,817	8,005	30,942	14,810	33,409
21	1,072	-9,318	10,646	0,905	1,878
22	10,774	0,557	20,542	6,099	12,576
23	17,743	10,172	30,402	12,866	26,745
24	1,044	-9,687	10,202	0,913	1,907
25	0,805	-10,186	10,159	0,563	1,186
26	8,618	3,389	23,085	7,304	15,069
27	33,591	17,550	37,539	22,052	45,809
28	28,361	13,806	33,683	18,804	38,962
29	5,732	-5,890	14,413	3,010	6,346
30	8,615	3,158	22,836	7,195	14,836

Крім того ширини довірчих інтервалів нелінійної регресії виявилися меншими у порівнянні з відповідними інтервалами лінійної регресії для 26 Web-застосунків (1, 2, 4-10, 12-26, 29 і 30). Причому найбільша різниця (5554%) виявилася для застосунку з номером 14 з невеликою кількістю строк коду – 470 (0,470 KLOC), а найменша різниця (1,4%) – для застосунку з номером 28 з 28361 строками коду (28,361 KLOC).

Отримані результати свідчать про те, що для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку

Laravel, потрібно використовувати нелінійні регресійні моделі на основі нормалізуючих перетворень.

1.2. Засоби реалізації програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel

Для розробки програмного забезпечення необхідно визначитись з мовою програмування, якою можливо реалізувати це ПЗ.

Для вирішення цього можна використовувати любі мови програмування, але оскільки програма виконує складні математичні обчислення, бажано щоб мова програмування та середовище розробки дозволяло легко працювати з ними. Цим умовам відповідає пакет прикладних програм для вирішення задач моделювання і обчислень MATLAB, з вбудованою однойменною мовою програмування та Scilab, з вбудованою Sci-мовою.

Мова MATLAB (<https://www.mathworks.com/>) є високорівневою інтерпретованою мовою програмування. Включає засновані на матрицях структури даних, широкий спектр функцій, інтегроване середовище розробки, об'єктно-орієнтовані можливості і інтерфейси до програм, написаних на інших мовах програмування.

Переваги – багата вбудована функціональність, працює на більшості ОС. Недоліки цього рішення – воно платне та потребує багато обчислювальних ресурсів.

Scilab є програмним пакетом для наукових обчислень, що вільно розповсюджується. Він є певною альтернативою комерційним пакетам моделювання MATLAB/Simulink та MATRIXx/SystemBuild [31, 32]. Остання версія Scilab – це Scilab 6.0.0 (<http://www.scilab.org>).

Scilab має схожу з MATLAB мову програмування – Sci-мову [31, 32]. Sci-мова є високорівневою структурованою мовою програмування. Дозволяє

працювати з елементарними і великим числом спеціальних функцій, має потужні засоби роботи з матрицями, поліномами, також проводити чисельні обчислення і рішення задач лінійної алгебри, оптимізації і моделювання, потужні статистичні функції, а також засіб для побудови та роботи з графіками.

Переваги – безкоштовне для комерційного використання, достатній вбудований функціонал, не потребує багато обчислювальних ресурсів, працює на більшості ОС. Для розробки програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel недоліків не виявлено.

Отже будемо використовувати середовище Scilab для розробки програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.

Так як Scilab вже містить вбудоване середовище програмування та побудови інтерфейсів, то для складання програми за допомогою обраної мови не має потреби використовувати різноманітні середовища розробки, які дозволяють значно зменшити витрати часу програміста щодо проектування програмної системи, проектування інтерфейсу користувача, написання програмного коду програми.

Для реалізації віконного інтерфейсу користувача та математичних алгоритмів обробки інформації добре підійде ця мова програмування, тому що вона добре документована та широко розповсюджена для наукових обчислень, а також має дуже схожу з MATLAB мову програмування. Це є також важливим моментом, оскільки, як правило, розроблене ПЗ необхідно підтримувати з часом, а іноді розробника (автора) ПЗ в момент необхідності внесення змін до програми знайти неможливо, тому виникає необхідність пошуку іншого програміста для доробки програми. Звичайно такого спеціаліста важче знайти, якщо програмне забезпечення було розроблено маловідомими засобами.

Виходячи з цього можна зробити висновок, що ця Sci-мова добре підійде для розробки програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.

1.3 Постановка задачі

Виходячи з аналізу існуючих методів та моделей для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel, сформулюємо постановку задачі. Потрібно створити програму для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel, 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 Web-застосунків: перша метрика Y включає фактичний розмір ПЗ в тисячах строк коду (KLOC), друга X_1 , третя X_2 та четверта X_3 метрики визначають відповідно загальну кількість класів, середню кількість методів та суму середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти.

2) Вводити з інтерфейсу програми довірчу ймовірність, три метрики діаграми класів, за якими потрібно оцінити розмір ПЗ Web-застосунку в тисячах строк коду: друга X_1 , третя X_2 та четверта X_3 метрики визначають відповідно загальну кількість класів, середню кількість методів та суму середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти.

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

4) Знаходити значення коефіцієнтів лінійної регресійної моделі за нормалізованими емпіричними даними.

5) Здійснювати побудову нелінійної регресійної моделі.

6) Обчислювати значення коефіцієнту детермінації R^2 .

7) Здійснювати побудову інтервалу передбачення та інтервалу прогнозування нелінійної регресії.

8) Виводити на екран нелінійну регресію, довірчий інтервал та інтервал прогнозування нелінійної регресії розміру Web-застосунку в тисячах строк коду, значення коефіцієнту детермінації R^2 .

9) Виводити у текстовий файл результати оцінювання розміру Web-застосунку в тисячах строк коду.

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.

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

2 РОЗРОБКА ПРОЕКТУ ПРОГРАМИ ДЛЯ ОЦІНЮВАННЯ КІЛЬКОСТІ СТРОК КОДУ WEB-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ ЗА ДОПОМОГОЮ ФРЕЙМВОРКУ LARAVEL

2.1 Розробка ескізного проекту програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel

В ескізному проекті було виявлено варіанти використання та побудовано їх сценарії, розроблено зовнішню специфікацію програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel 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 2.1.

Таблиця 2.1 Виявлені варіанти використання

Основні актори	Найменування	Формулювання
1	2	3
Користувач	Ввести дані метрик ПЗ Web-застосунків	Вводить дані про фактичний розмір Web-застосунків в тисячах строк коду, загальну кількість класів, середню кількість методів та суму середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти.
Користувач	Ввести три метрики діаграми класів	Вводить три метрики діаграми класів, за якими потрібно оцінити розмір ПЗ Web-застосунку в тисячах строк коду: загальну кількість класів, середню кількість методів та суму середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти.

Продовження табл.2.1

1	2	3
Користувач	Ввести довірчу ймовірність	Вводить довірчу ймовірність, для якої потрібно оцінити довірчий інтервал нелінійної регресії кількості строк коду Web-застосунків.
Користувач	Нормалізувати дані	Виконує нормалізацію внесених даних
Користувач	Побудувати нелінійну регресійну модель	Будує нелінійну регресію для оцінювання середнього значення кількості строк коду Web-застосунків.
Користувач	Отримати довірчий інтервал та інтервал прогнозування нелінійної регресії кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel	Отримує довірчий інтервал та інтервал прогнозування нелінійної регресії для інтервального оцінювання вибіркового середнього кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel, із заданою ймовірністю

Діаграма варіантів використання програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel, зображена на рисунку 2.1.

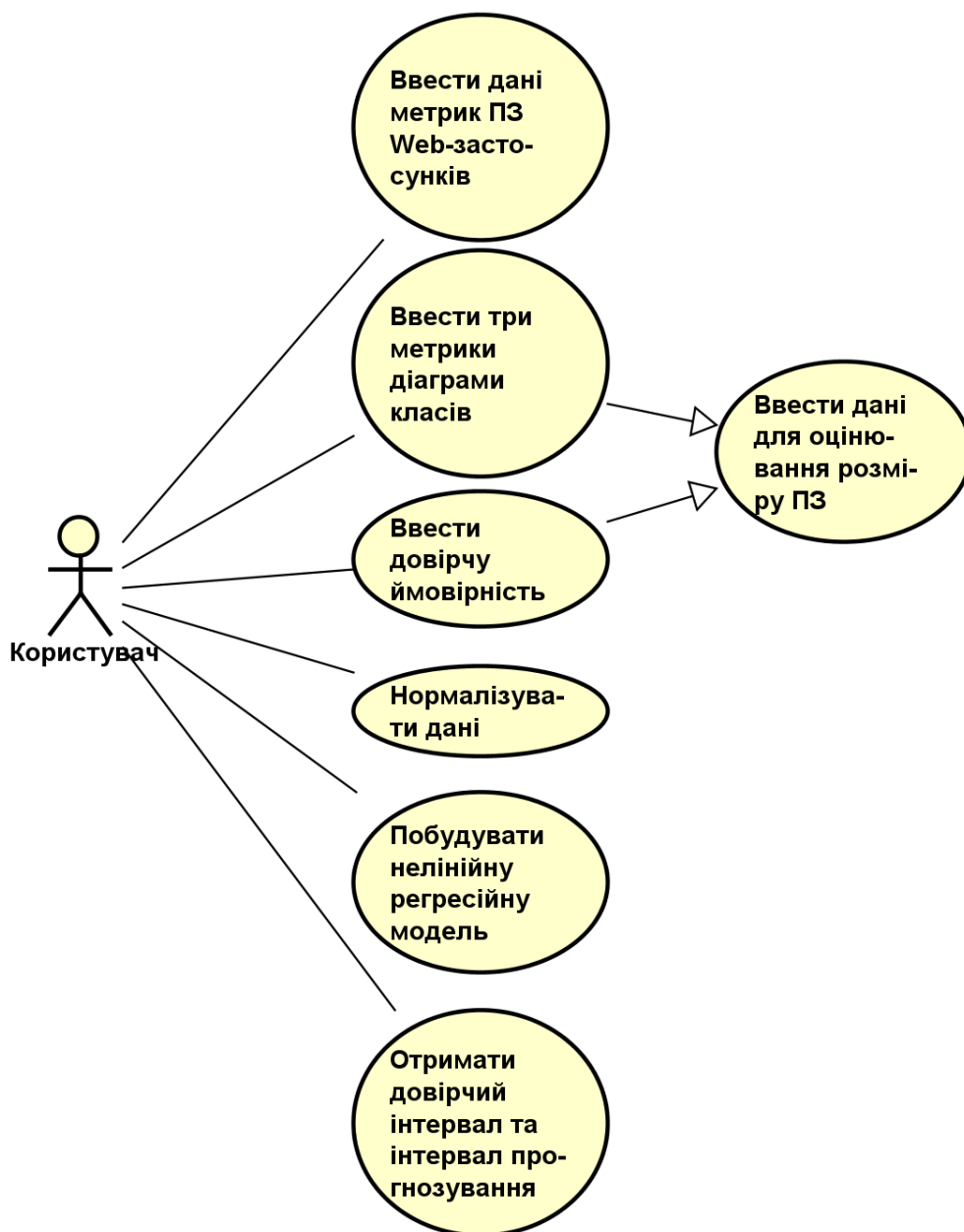


Рисунок 2.1 – Діаграма варіантів використання ПЗ для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel

Сценарії варіантів використання зображені у вигляді діаграми діяльності на рисунку 2.2.

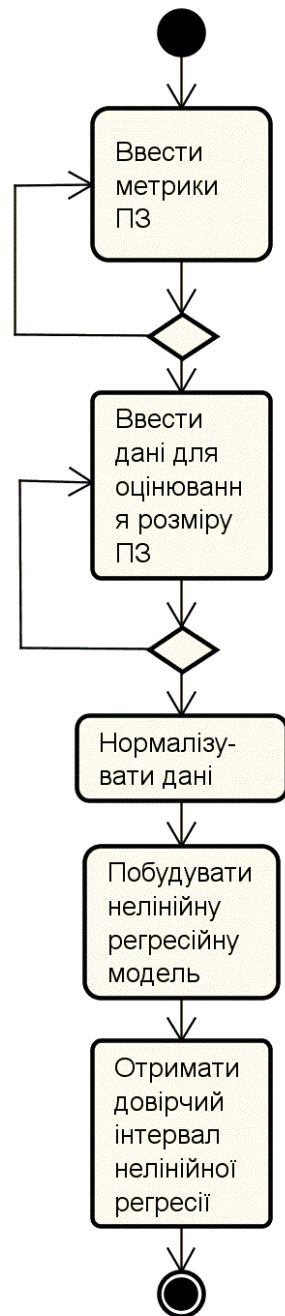


Рисунок 2.2 – Діаграма діяльності ПЗ для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel

Зовнішня специфікація програми.

Назва програми: WebSoftSizeEst.

Функції, які виконує програма:

1) Вводити з текстового файлу емпіричні дані з метрик Web-застосунків (значення випадкових величин): перша метрика Y включає фактичний розмір ПЗ в тисячах строк коду (KLOC), друга X_1 , третя X_2 та четверта X_3 метрики визначають відповідно загальну кількість класів, середню кількість методів та суму середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти.

2) Вводити з інтерфейсу програми довірчу ймовірність, три метрики діаграми класів, за якими потрібно оцінити розмір ПЗ Web-застосунку в тисячах строк коду: загальну кількість класів, середню кількість методів та суму середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти.

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

4) Знаходити значення коефіцієнтів лінійної регресійної моделі за нормалізованими емпіричними даними.

5) Здійснювати побудову нелінійної регресійної моделі.

6) Обчислювати значення коефіцієнту детермінації R^2 .

7) Здійснювати побудову довірчого інтервалу та інтервалу прогнозування нелінійної регресії.

8) Виводити на екран нелінійну регресію, довірчий інтервал та інтервал прогнозування нелінійної регресії розміру Web-застосунку в тисячах строк коду, значення коефіцієнту детермінації R^2 .

9) Виводити у текстовий файл результати оцінювання розміру Web-застосунку у тисячах строк коду.

Вхідні дані:

- емпіричні дані з метрик Web-застосунків (значення випадкових величин): перша метрика Y включає фактичний розмір програмного забезпечення в тисячах строк коду (KLOC), друга X_1 , третя X_2 та четверта X_3 метрики визначають відповідно загальну кількість класів, середню

кількість методів та суму середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти;

- значення довірчої ймовірності, трьох метрик діаграми класів, за якими потрібно оцінити розмір Web-застосунку в тисячах строк коду: друга X_1 , третя X_2 та четверта X_3 метрики визначають відповідно загальну кількість класів, середню кількість методів та суму середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти.

Вихідні дані:

- значення нелінійної регресії, границь довірчого інтервалу та інтервалу прогнозування нелінійної регресії кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.

- значення коефіцієнту детермінації.

Зовнішні ефекти: поява графічного вікна для вводу даних, управління виводом даних, графічного вікна для виводу даних, нелінійної регресії, довірчого інтервалу та інтервалу прогнозування нелінійної регресії кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel, вихід з програми (закриття всіх графічних вікон).

Для проектування інтерфейсу програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel представимо стани програми при виконанні у вигляді відповідної діаграми станів. Діаграма станів програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel, наведена на рисунку 2.3.

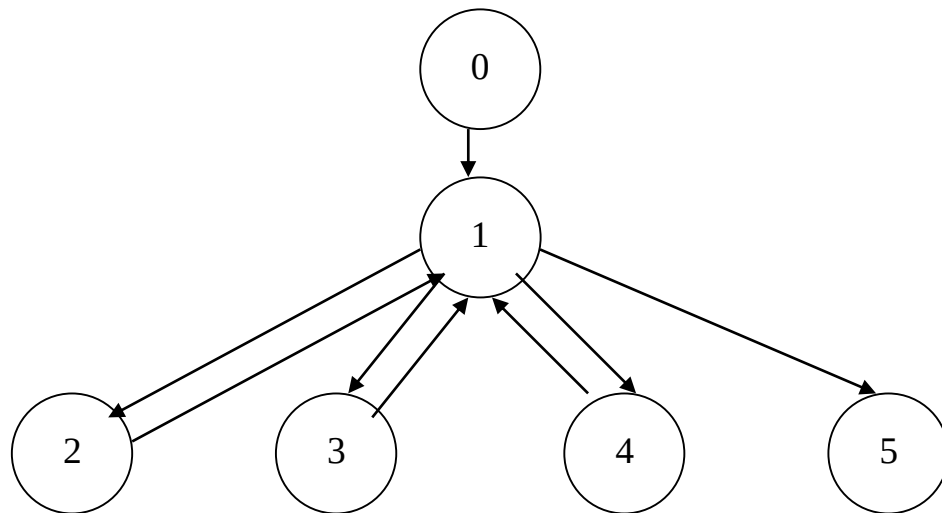


Рисунок 2.3 – Діаграма станів програми для оцінювання розміру Web-застосунків

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 знаходження нелінійної регресії, 4 – графічне вікно з результатами побудови довірчого інтервалу та інтервал прогнозування нелінійної регресії кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel, 5 – кінцевий стан.

Далі виконуємо розробку технічного проекту програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel

2.2 Розробка технічного проекту програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel

В технічному проєкті було спроектовано програму для оцінювання

кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel, на основі покрокової деталізації зверху вниз, розроблено специфікації модулів програми, побудовано алгоритми модулів програми.

Програма складається з таких модулів: головного модуля (WebSoftSizeEst), модуля для введення емпіричних даних з файлу та їх візуалізації (DataInput), модуля для побудови нелінійної регресії та довірчого інтервалу нелінійної регресії кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel (WebSoftSizeConfIntEst), модуля для закриття графічних вікон (W_Close).

Специфікація головного модуля.

Назва модуля: WebSoftSizeEst.

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 фактичного розміру Web-застосунку та введення його значення у тисячах строк коду.
- 5) Виклик функції для визначення нелінійної регресії та довірчого інтервалу нелінійної регресії кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.
- 6) Виклик функції для закриття графічних вікон.

Вхідні дані:

- емпіричні дані з метрик Web-застосунків з відкритим кодом: перша метрика Y включає фактичний розмір програмного забезпечення в тисячах строк коду (KLOC), друга X_1 , третя X_2 та четверта X_3 метрики визначають відповідно загальну кількість класів, середню кількість методів та суму середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти;
- значення довірчої ймовірності, трьох метрик діаграми класів, за

якими потрібно оцінити розмір Web-застосунку в тисячах строк коду: загальну кількість класів, середню кількість методів та суму середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти.

Вихідні дані:

- значення нелінійної регресії, границь довірчого інтервалу та інтервалу прогнозування нелінійної регресії кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.

- значення коефіцієнту детермінації.

Зовнішні ефекти: поява графічного вікна для вводу даних, управління виводом даних, побудови довірчого інтервалу та інтервалу прогнозування нелінійної регресії кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel, 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 (значень випадкових величини) 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 визначення нелінійної регресії та довірчого інтервалу нелінійної регресії кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.

Назва модуля: WebSoftSizeConfIntEst.

Функції, які виконує модуль:

1) Визначає нелінійну регресію, довірчий інтервал та інтервалу прогнозування нелінійної регресії кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.

2) Обчислює значення коефіцієнту детермінації для нелінійної регресії.

Вхідні дані:

- двомірні масиви значень випадкових величини (емпіричних даних);
- кількість значень випадкових величин (емпіричних даних).

Вихідні дані:

- значення нелінійної регресії, нижньої, верхньої границь довірчого інтервалу та інтервалу прогнозування нелінійної регресії кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel;
- значення коефіцієнту детермінації для нелінійної регресії.

Зовнішні ефекти: з'являються результати нелінійної регресії, нижньої та верхньої границь довірчого інтервалу та інтервалу прогнозування нелінійної регресії кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel, та виводиться значення коефіцієнту детермінації для нелінійної регресії.

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.

2.3 Розробка робочого проекту програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel

В робочому проекті було розроблено програму для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel, здійснено тестування програми, розроблено програмну документацію, здійснено випробування програми.

Спочатку scі-мовою системи моделювання Scilab було виконано кодування алгоритмів, що були розроблені на стадії технічного проекту. Текст програми наведений в додатку Б. Приклад кодування модуля WebSoftSizeEst, що здійснює оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel, scі-мовою для системи моделювання Scilab наведено нижче.

```
//Функція Оцінювання кількості строк коду Web-застосунків, що
створюються за допомогою фреймворку Laravel
function WebSoftSizeEst(x, yx, 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_text7, '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; 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(x1i)-z1A; zi(2)=log10(x2i)-z2A; zi(3)=log10(x3i)-z3A;
ziT=zi';
ziSzz_1=ziT*Szz_1;
result=ziSzz_1*zi;

Df=n-4;
alpha=1-Pconf;
P=1-alpha/2;
//P=0.975;
Q=1-P;

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

b0=-1.74968;

```

```

b1=1.07738;
b2=0.76106;
b3=-0.19497;

Szyzyn4=0; Syy=0;
for i=1:n
    Szyzyn4=Szyzyn4+...
        (b0+b1*log10(x(i,1))+b2*log10(x(i,2))+b3*log10(x(i,3))-
log10(yx(i,1)))^2;

    Syy=Syy+(yx(i,1)-10^b0*x(i,1)^b1*x(i,2)^b2*x(i,3)^b3)^2;
end;
Szyzyn4=Szyzyn4/(n-4)
Delta_z=T*(Szyzyn4*(1/n+result))^0.5;
Delta_z_pr=T*(Szyzyn4*(1+1/n+result))^0.5;

r_2=1-Syy/SyA;

zy_result=b0+b1*log10(x1i)+b2*log10(x2i)+b3*log10(x3i);

zyLBC=zy_result-Delta_z; zyUBC=zy_result+Delta_z;

zyLBPr=zy_result-Delta_z_pr; zyUBPr=zy_result+Delta_z_pr;

y_result=10^zy_result;
yLBC=10^zyLBC; yUBC=10^zyUBC;
yLBPr=10^zyLBPr; yUBPr=10^zyUBPr;

file_PC5=fopen('D:\Data_F\Web_Laravel_30_data5.txt','w+');
//for i=1:3
    fprintf(file_PC5, "%5.6f ", yUBC);
//end
fclose(file_PC5);

file_PC4=fopen('D:\Data_F\Web_Laravel_30_data4.txt','w+');
for i = 1:3
    for j = 1:3 fprintf(file_PC4, "%5.6f ", Szz_1(i,j)); end;
    fprintf(file_PC4, "\n");
end;
fclose(file_PC4);

set(textresult,'string', sprintf("=%1.4f",y_result));
set(textresult2,'string', sprintf("=%1.4f",yLBC));
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 (get(audio_bikv_x,'value')==0)

else
    yfact=eval(get(edit_text12,'string'));

```

```

MRE=abs((yfact-y_result)/yfact);

textresult5=uicontrol(f,'style','text','string','', 'position',...
.
    [500,120,160,20]);
set(textresult5,'string',...
sprintf("Відносна похибка MRE=%1.4f",MRE));
end;

endfunction

```

Далі в середовищі Scilab 6.0.0 було створено sce-файл розміром 12 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.2.4.

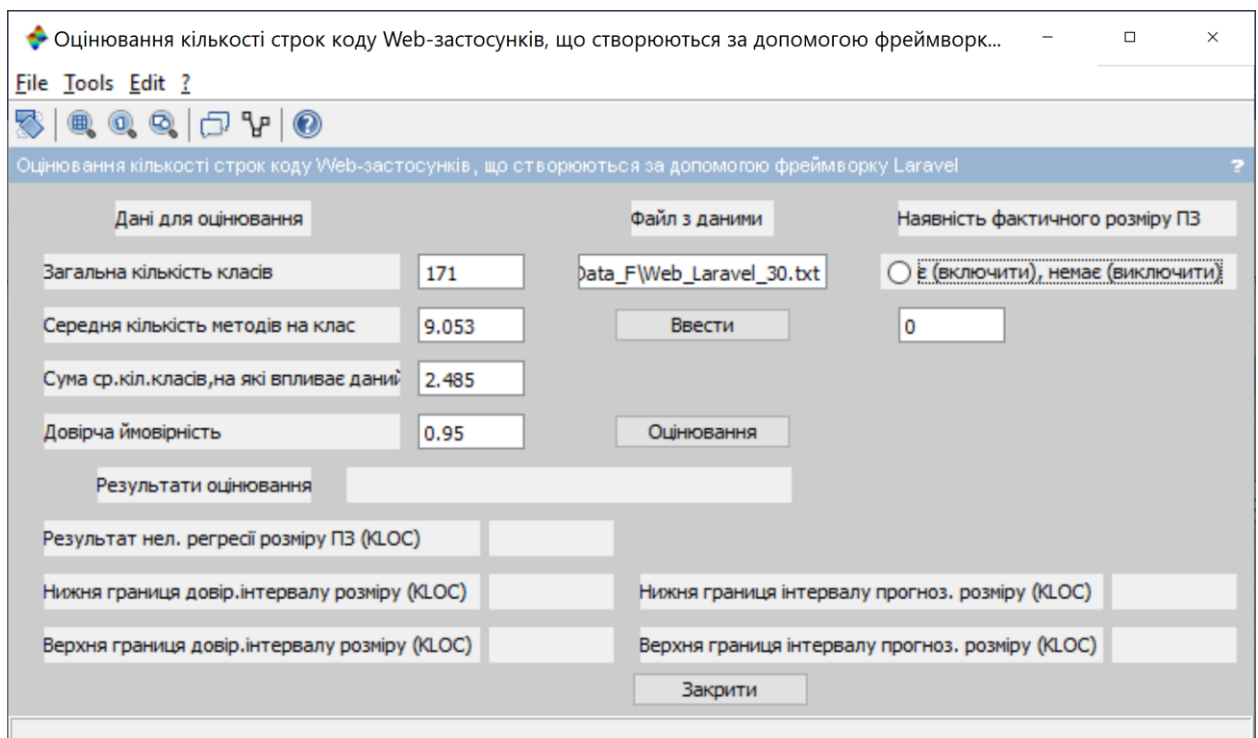


Рисунок 2.4. – Графічне вікно програми із інтерфейсними компонентами

Було проведено тестування окремих модулів. У якості прикладу наведемо результати тестування для модуля оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel для емпіричних даних – кількості строк коду (у тисячах строк коду) 30 Web-застосунків в залежності від загальної кількості класів, загальної кількості зв'язків між класами та суми середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти. Було побудовано нелінійне рівняння регресії на основі нормалізуючого перетворення у вигляді десяткового логарифму. Результати візуалізації емпіричних даних про кількість строк коду (у тисячах строк коду) 30 Web-застосунків в залежності від загальної кількості класів наведені на рис.2.5.

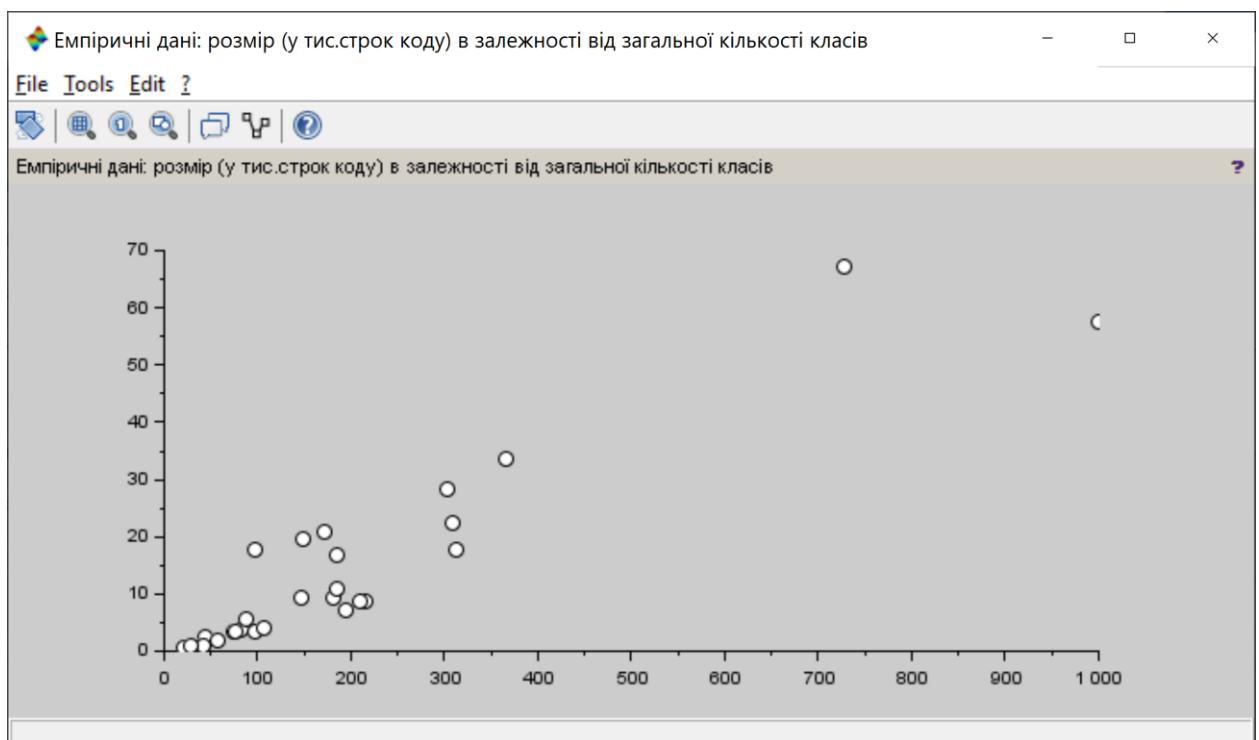


Рисунок 2.5. – Емпіричні дані про кількість строк коду (у тисячах строк коду) 30 Web-застосунків в залежності від загальної кількості класів

Результати візуалізації емпіричних даних про кількість строк коду (у

тисячах строк коду) 30 Web-застосунків в залежності від середньої кількості методів на клас наведені на рис.2.6.

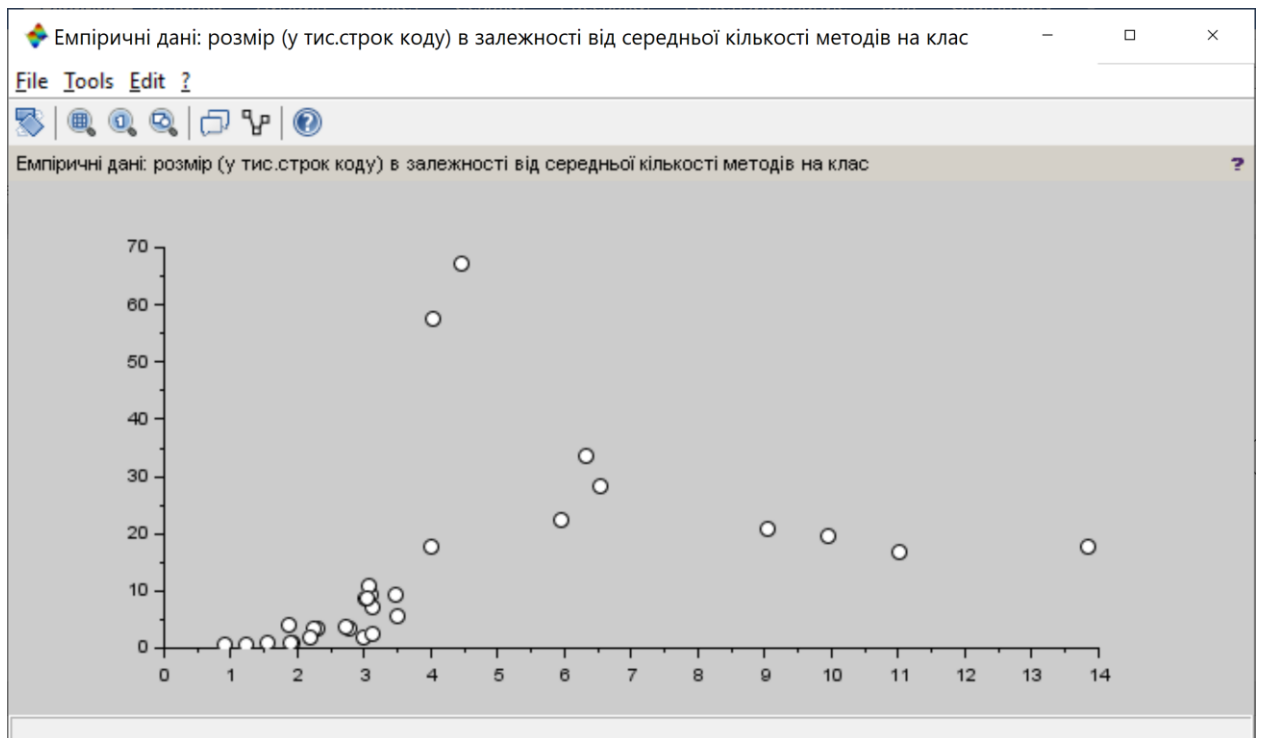


Рисунок 2.6. – Емпіричні дані про кількість строк коду (у тисячах строк коду) 30 Web-застосунків в залежності від середньої кількості методів на клас

Результати візуалізації емпіричних даних про кількість строк коду (у тисячах строк коду) 30 Web-застосунків в залежності від суми середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти наведені на рис.2.7.

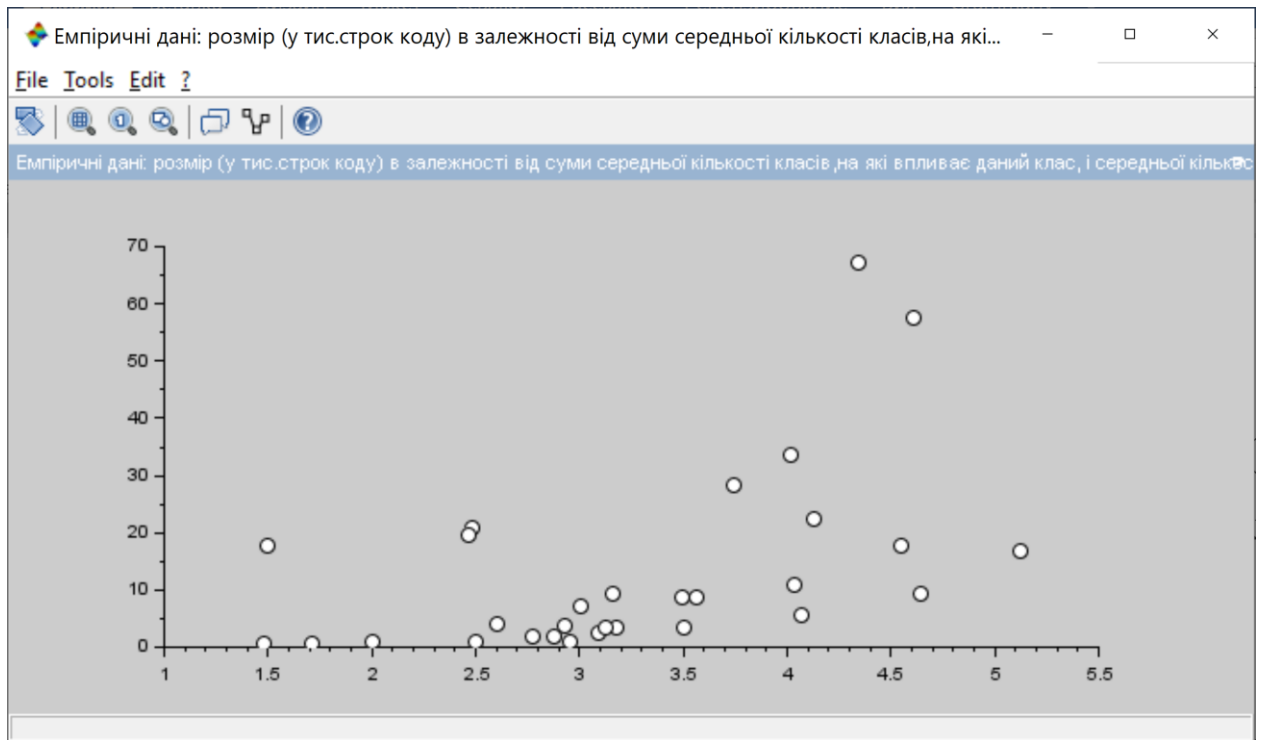


Рисунок 2.7. – Емпіричні дані про кількість строк коду (у тисячах строк коду) 30 Web-застосунків в залежності від суми середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти

Результати оцінювання значень нелінійної регресії, нижньої і верхньої границь довірчого інтервалу нелінійної регресії кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel, та результати обчислення коефіцієнту детермінації R^2 наведені на рис.2.8. На рис.2.8 також наведено значення відносної похибки (MRE) результату оцінювання значень нелінійної регресії в порівнянні з фактичним розміром програмного забезпечення Web-застосунку для зазначених метрик.

Оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворк...

File Tools Edit ?

Оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel

Дані для оцінювання		Файл з даними	Наявність фактичного розміру ПЗ
Загальна кількість класів	25	Data_F\Web_Laravel_30.txt	☒ є (включити), немає (виключити)
Середня кількість методів на клас	1.08	Ввести	0.601
Сума ср.кіл.класів,на які впливає даний	1.76		
Довірча ймовірність	0.95	Оцінювання	
Результати оцінювання		Коефіцієнт детермінації $R^2=0.9461$	
Результат нел. регресії розміру ПЗ (KLOC)	=0.5420	Відносна похибка MRE=0.0982	
Нижня границя довір.інтервалу розміру (KLOC)	=0.4630	Нижня границя інтервалу прогноз. розміру (KLOC)	=0.3689
Верхня границя довір.інтервалу розміру (KLOC)	=0.6345	Верхня границя інтервалу прогноз. розміру (KLOC)	=0.7963
Закрити			

Рисунок 2.8. – Результати оцінювання значень нелінійної регресії, нижньої і верхньої границь довірчого інтервалу нелінійної регресії розміру Web-застосунків

Отримані результати практично співпали з результатами, що отримані іншими авторами. Це свідчить про працездатність розробленої програми.

З РЕЗУЛЬТАТИ РОЗРОБКИ

В результаті розробки було створено програму для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel для системи моделювання Scilab 6.0.0. Назва програми: WebSoftSizeEst. В середовищі Scilab 6.0.0 було створено sce-файл WebSoftSizeEst розміром 12 КБ. Проведено тестування програми.

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. Результати випробувань програми наведені на рис. 3.1-3.5. При випробуванні програми вводилися із текстового файлу наступні тестові дані:

```
20.816 171 9.053 2.485
9.482 180 3.100 4.644
57.454 1000 4.039 4.612
19.481 148 9.939 2.466
3.532 75 2.773 3.173
17.888 98 13.847 1.500
3.825 81 2.728 2.926
22.438 308 5.948 4.127
1.881 44 3.000 2.773
2.364 44 3.114 3.091
67.055 727 4.455 4.344
9.305 146 3.466 3.158
3.364 97 2.299 3.124
0.47 23 0.913 1.478
3.391 76 2.250 3.500
4.069 107 1.879 2.607
7.241 194 3.108 3.010
```



```

0.515 21 1.238 1.714
1.771 57 2.175 2.877
16.817 185 11.000 5.119
1.072 40 1.925 2.500
10.774 184 3.065 4.038
17.743 313 3.994 4.550
1.044 42 1.905 2.952
0.805 29 1.552 2.000
8.618 215 3.019 3.563
33.591 365 6.312 4.014
28.361 303 6.535 3.743
5.732 88 3.500 4.068
8.615 210 3.043 3.490

```

Після запуску програми відкривається графічне вікно «Оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel», головного модуля програми (рис. 3.1). Далі у вікні редагування «Файл з даними» цього графічного вікна потрібно відредагувати ім'я текстового файлу з початковими емпіричними даними, що будуть використовуватися для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.

Рисунок 3.1 – Графічне вікно головного модуля програми

Крім того у відповідних вікнах редагування потрібно відредагувати: загальну кількість класів, середню кількість методів на клас, суму середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти, довірчу ймовірність.

Після натискання на командну кнопку «Ввести» з текстового файлу вводяться дані (фактичний розмір ПЗ в тисячах строк коду, загальна кількість класів, середня кількість методів на клас та сума середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти), які виводяться у окремих графічних вікнах (рис. 3.2 - 3.4).

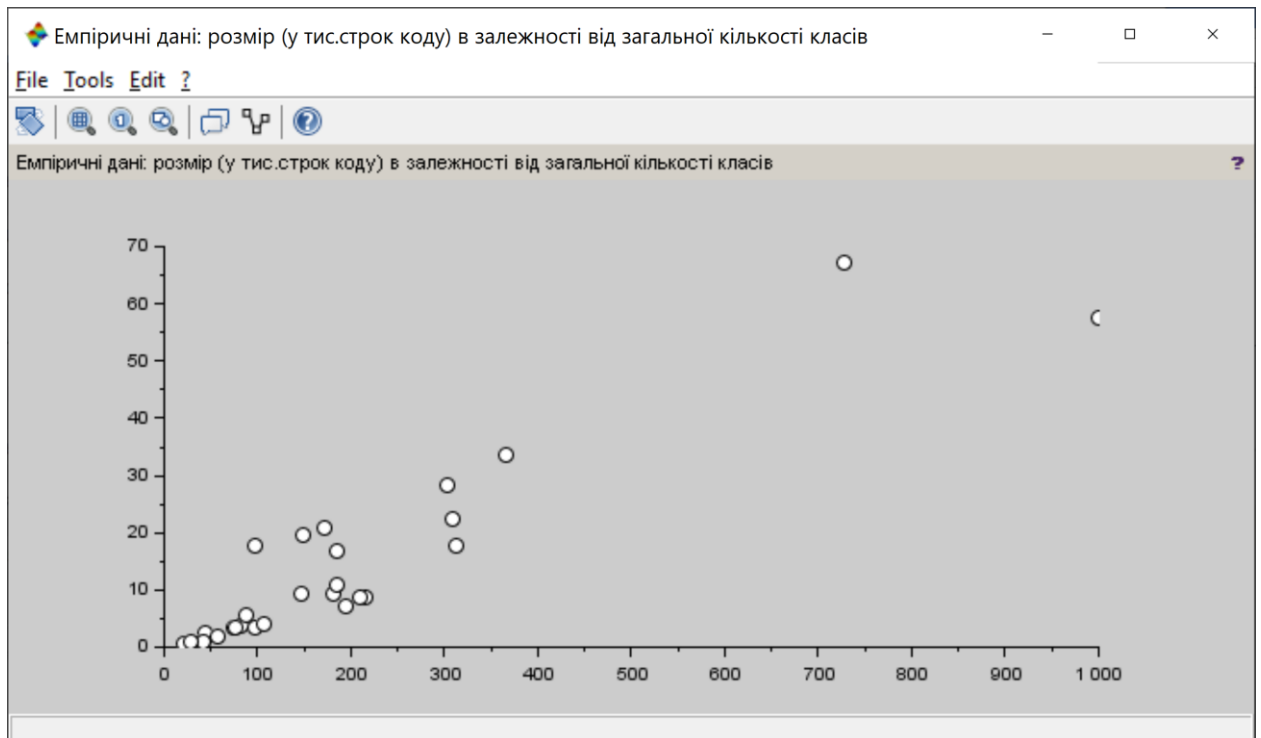


Рисунок 3.2 – Емпіричні дані про кількість строк коду (у тисячах строк коду)
30 Web-застосунків в залежності від загальної кількості класів

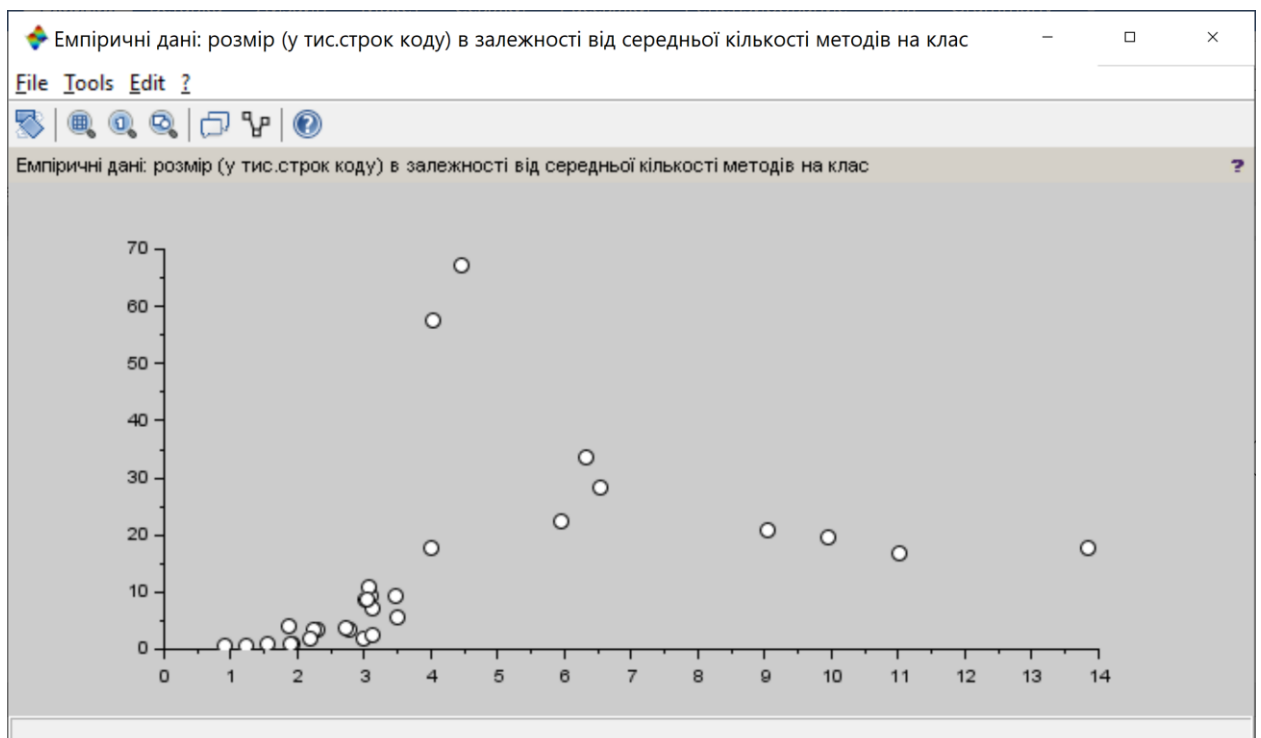


Рисунок 3.3 – Емпіричні дані про кількість строк коду (у тисячах строк коду)
30 Web-застосунків в залежності від середньої кількості методів на клас

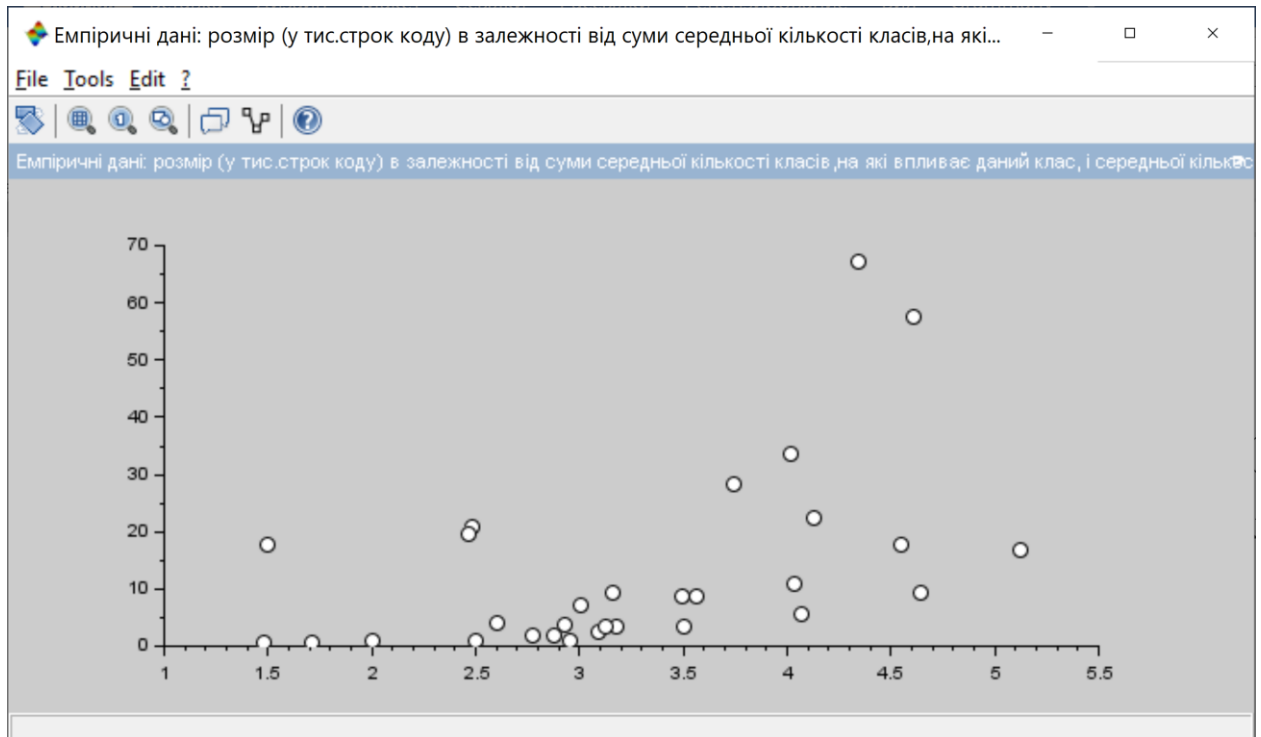


Рисунок 3.4 – Емпіричні дані про кількість строк коду (у тисячах строк коду) 30 Web-застосунків в залежності від суми середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти

Потім у графічному вікні «Оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel» головного модуля програми (рис. 3.1) потрібно натиснути на командну кнопку «Оцінювання». Результати роботи програми після натискання на командну кнопку «Оцінювання» у разі, коли радіо кнопка виключена, наведені на рис.3.5.

Оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворк...

File Tools Edit ?

Оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel

Дані для оцінювання		Файл з даними	Наявність фактичного розміру ПЗ
Загальна кількість класів	171	Data_F\Web_Laravel_30.txt	☐ є (включити), немає (виключити)
Середня кількість методів на клас	9.053	Ввести	0
Сума ср.кіл.класів,на які впливає дані	2.485		
Довірча ймовірність	0.95	Оцінювання	

Результати оцінювання Коефіцієнт детермінації $R^2=0.9461$

Результат нел. регресії розміру ПЗ (KLOC)	=20.2864		
Нижня границя довір.інтервалу розміру (KLOC)	=17.5550	Нижня границя інтервалу прогноз. розміру (KLOC)	=13.8784
Верхня границя довір.інтервалу розміру (KLOC)	=23.4427	Верхня границя інтервалу прогноз. розміру (KLOC)	=29.6531

Закрити

Рисунок 3.5 – Результати роботи програми після натискання на командну кнопку «Оцінювання» у разі, коли радіо кнопка виключена

Результати роботи програми після натискання на командну кнопку «Оцінювання» у разі, коли радіо кнопка включена та введений фактичний розмір ПЗ – 20,816 тис. строк коду, наведені на рис.3.6.

Оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворк...

File Tools Edit ?

Оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel

Дані для оцінювання		Файл з даними	Наявність фактичного розміру ПЗ
Загальна кількість класів	171	Data_F\Web_Laravel_30.txt	☒ є (включити), немає (виключити)
Середня кількість методів на клас	9.053	Ввести	20.816
Сума ср.кіл.класів,на які впливає дані	2.485		
Довірча ймовірність	0.95	Оцінювання	
Результати оцінювання		Коефіцієнт детермінації $R^2=0.9461$	
Результат нел. регресії розміру ПЗ (KLOC)		=20.2864	
Нижня границя довір.інтервалу розміру (KLOC)		=17.5550	
Верхня границя довір.інтервалу розміру (KLOC)		=23.4427	
		Нижня границя інтервалу прогноз. розміру (KLOC) =13.8784	
		Верхня границя інтервалу прогноз. розміру (KLOC) =29.6531	
Закрити			

Рисунок 3.6 – Результати роботи програми після натискання на командну кнопку «Оцінювання» у разі, коли радіо кнопка включена та введений фактичний розмір ПЗ – 20,816 тис. строк коду

Результати роботи програми після натискання на командну кнопку «Оцінювання» у разі, коли радіо кнопка включена, та введене нове значення довірчої ймовірності – 0,99, наведені на рис.3.7.

Оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворк...

File Tools Edit ?

Оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel

Дані для оцінювання		Файл з даними	Наявність фактичного розміру ПЗ
Загальна кількість класів	171	Data_F\Web_Laravel_30.txt	☒ є (включити), немає (виключити)
Середня кількість методів на клас	9.053	Ввести	20.816
Сума ср.кіл.класів,на які впливає дані	2.485		
Довірча ймовірність	0.99	Оцінювання	

Результати оцінювання

Коефіцієнт детермінації $R^2=0.9461$

Відносна похибка MRE=0.0254

Результат нел. регресії розміру ПЗ (KLOC)	=20.2864		
Нижня границя довір.інтервалу розміру (KLOC)	=16.6842	Нижня границя інтервалу прогноз. розміру (KLOC)	=12.1433
Верхня границя довір.інтервалу розміру (KLOC)	=24.6663	Верхня границя інтервалу прогноз. розміру (KLOC)	=33.8902

Закрити

Рисунок 3.7 – Результати роботи програми після натискання на командну кнопку «Оцінювання» у разі, коли радіо кнопка включена, та введене нове значення довірчої ймовірності – 0,99

За результатами випробувань зроблено висновок про працездатність розробленої програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel, для системи моделювання Scilab 6.0.0.

4 ОХОРОНА ПРАЦІ

4.1 Вступ

Науково-технічний прогрес вніс серйозні зміни в умови виробничої діяльності робітників розумової праці. Їх праця стала більш інтенсивним, напруженим, які вимагають значних витрат розумової, емоційної і фізичної енергії. Це зажадало комплексного рішення проблем ергономіки, гігієни і організації праці, регламентації режимів праці та відпочинку.

В даний час комп'ютерна техніка широко застосовується у всіх областях діяльності людини. При роботі з комп'ютером людина піддається дії ряду небезпечних і шкідливих виробничих факторів: електромагнітних полів (діапазон радіочастот: ВЧ, УВЧ і СВЧ), інфрачервоного і іонізуючого випромінювань, шуму і вібрації, статичної електрики та ін.

Робота з комп'ютером характеризується значною розумовою напругою і нервово-емоційним навантаженням операторів, високою напруженістю зорової роботи і достатньо великим навантаженням на м'язи рук при роботі з клавіатурою персонального комп'ютера (ПК). Велике значення має раціональна конструкція і розташування елементів робочого місця, що важливо для підтримки оптимальної робочої пози людини-оператора.

У процесі роботи з комп'ютером необхідно дотримувати правильний режим праці та відпочинку. В іншому випадку у персоналу наголошуються значна напруга зорового апарату з появою скарг на незадоволеність роботою, головні болі, дратівливість, порушення сну, втому і хворобливі відчуття в очах, в поясниці, в області шиї і руках.

4.2 Аналіз шкідливих та небезпечних факторів при роботі з персональними комп'ютерами

Основним фактором, що впливає на продуктивність праці людей, що працюють з ПК, є комфортні і безпечні умови праці.

Дотримання нижче викладених рекомендацій і правильна організація робочого місця за комп'ютером допоможе перетворити роботу в задоволення і уникнути негативного впливу комп'ютера на здоров'я.

Площа робочого місця користувача ПК повинна складати не менше 6 м². У приміщеннях повинна проводитися щоденне вологе прибирання і систематичне провітрювання після кожної години роботи. Шумне обладнання (друкують устрою, сканери, сервери і тому подібні), рівні шуму якого перевищують нормативні, повинне розміщуватися поза робочим місцем користувача.

Робочі столи слід розміщувати таким чином, щоб монітори були орієнтовані бічною стороною до світлових прорізів, щоб природне світло падало переважно ліворуч.

Конструкція робочого столу повинна забезпечувати оптимальне розміщення на робочій поверхні використовуваного обладнання. Висота робочої поверхні столу повинна бути 725 мм, робоча поверхня стола повинна мати ширину 800..1400 мм і глибину 800..1000 мм. Робочий стіл повинен мати простір для ніг висотою не менше 600 мм, шириною – не менше 500 мм, глибиною на рівні колін - не менше 450 мм і на рівні витягнутих ніг - не менше 650 мм.

Конструкція робочого стільця або крісла повинна забезпечувати підтримку раціональної робочої пози працівника і дозволяти змінювати позу з метою зниження статичного напруження м'язів шийно-плечової області і спини. Робочий стілець або крісло повинні бути підйомно-поворотним, регульованим по висоті і кутам нахилу сидіння і спинки, а також відстані

спинки від переднього краю сидіння, при цьому регулювання кожного параметра повинна бути незалежною, легко здійснюваною плюс надійну фіксацію.

Клавіатуру слід розташовувати на поверхні столу на відстані 100..300 мм від краю, зверненого до користувача, або на спеціальній поверхні, відокремленої від основної стільниці. Екран монітора повинен знаходитися від очей користувача на відстані 600...700 мм, але не ближче 500 мм.

Вимоги безпеки перед початком роботи Перед початком роботи з ПК працівник зобов'язаний:

- 1) Провірити робоче приміщення;
- 2) Перевірити:
 - Стійкість положення обладнання на робочому столі;
 - Відсутність видимих пошкоджень обладнання, дискет в дисководі системного блоку;
 - Справність і цілісність живлять і сполучних кабелів, роз'ємів і штепсельних з'єднань, захисного заземлення (занулення);
 - Справність меблів.
- 3) Відрегулювати:
 - Положення стола, стільця (крісла), підставки для ніг, клавіатури, екрана монітора;
 - Освітленість на робочому місці. При необхідності включити місцеве освітлення;
- 4) Протерти поверхню екрану монітора, захисного фільтра (при його наявності) сухою м'якою тканинною серветкою;
- 5) Переконавшись у відсутності відображень на екрані монітора, зустрічного світлового потоку;
- 6) Включити обладнання ПК в електричну мережу, дотримуючись наступну послідовність: стабілізатор напруги (якщо він використовується), блок безперебійного живлення, периферійні пристрої (принтер, монітор, сканер та інші пристрої), системний блок.

Забороняється приступати до роботи при:

- 1) Вираженому тремтінні зображення на моніторі;
- 2) Виявленні несправності обладнання;
- 3) Наявності пошкоджених кабелів або проводів, роз'ємів, штепсельних з'єднань;
- 4) Відсутності або несправності захисного заземлення (занулення) обладнання.

4.3 Розрахунок пожежної безпеки на робочому місці користувача персонального комп'ютера

Для гасіння пожежі застосовуватиметься установка об'ємного пожежогасіння. Розміри приміщення: $A \times B \times H = 4 \times 5 \times 2,5$ м;

Кількість відкритих прорізів – 2 шт.

Площа відкритих прорізів по відношенню до площі огорожувальних конструкцій 4% та 5%. Оскільки сумарна площа відкритих прорізів не перевищує 15%, то можна застосувати об'ємне пожежогасіння. Обрано вогнегасний порошок загального призначення П-2АП (ТУ У6-057663662-001ТУ).

Загальний об'єм приміщення:

$$V = A * B * H = 4 * 5 * 2,5 = 50 \text{ куб.м.}$$

Для обраного порошку норма подачі для об'ємного гасіння 0,6 кг/куб.м.

Основна маса порошку становитиме:

$$M_1 = q_{v0} * V = 0,6 * 50 = 30 \text{ кг.}$$

Оскільки площа отворів менше ніж 5%, то для визначення додаткової маси вогнегасного порошку застосовуємо формулу:

$$M_2 = 2,5 * (S_1 + S_2) = 2,5 * (2 + 2,5) = 11,25 \text{ кг.}$$

Таким чином, загальна мінімальна кількість вогнегасного порошку, необхідного для захисту приміщення, буде складати:

$$M_{min} = M_1 + M_2 = 30 + 11,25 = 41,25 \text{ кг.}$$

Визначимо мінімальну витрату порошку:

$$G_{min} = \frac{M_{min}}{30} = \frac{41,25}{30} = 1,375 \frac{\text{кг}}{\text{с}}.$$

Визначимо мінімальну тривалість випуску вогнегасної речовини за формулою:

$$t_{min} = 0,67 q_{v0} * I_{v0}^{-1} = 0,67 * 0,6 * 0,02^{-1} = 20,1 \text{ с.}$$

За отриманими значеннями M_{min} , G_{min} , t_{min} обираємо установку порошкового пожежогасіння.

4.4 Вимоги безпеки при виконанні робіт

Роботу за екраном монітору слід періодично переривати на регламентовані перерви, які встановлюються для забезпечення працездатності та збереження здоров'я, або замінювати іншою роботою з метою скорочення робочого навантаження біля екрану.

Час регламентованих перерв протягом робочого дня (зміни) встановлюється в залежності від його (її) тривалості, виду і категорії трудової діяльності.

При 8-годинній робочій зміні і роботі з ПК регламентовані перерви встановлюються:

- 1) Для I категорії робіт через 2 години від початку робочої зміни і через 2 години після обідньої перерви тривалістю 15 хвилин кожен;
- 2) Для II категорії робіт через 2 години від початку робочої зміни і через 1,5-2 години після обідньої перерви тривалістю 15 хвилин кожен або тривалістю 10 хвилин кожен годину роботи;

3) Для III категорії робіт через 1,5-2 години від початку робочої зміни і через 1,5-2 години після обідньої перерви тривалістю 20 хвилин кожен або тривалістю 15 хвилин через кожну годину роботи.

Тривалість безперервної роботи з ПК без регламентованого перерви не повинна перевищувати 2 годин.

Під час регламентованих перерв для зниження нервово-емоційного напруження, стомлення зорового аналізатора, поліпшення функціонального стану, нервової, серцево-судинної, дихальної систем, а також м'язів плечового поясу, рук, спини, шиї та ніг доцільно виконувати комплекси фізичних вправ.

З метою зменшення негативного впливу монотонності необхідно застосовувати чергування операцій.

Не слід залишати обладнання включеним без нагляду. При необхідності припинення на деякий час роботи закриваються всі активні завдання та обладнання вимикається.

При роботі з ПК забороняється:

- 1) При включеному харчуванні торкатися до панелей з роз'ємами обладнання, роз'ємами живлять і сполучних кабелів, екрану монітора;
- 2) Проводити перемикання, відключення живлення під час виконання активної задачі;
- 3) Допускати потрапляння вологи на поверхню обладнання;
- 4) Включати сильно охолоджене (принесене з вулиці в зимовий час) обладнання;
- 5) Проводити самостійно розкриття і ремонт обладнання;
- 6) Витирати пил на включеному обладнанні.

Час регламентованих перерв протягом робочого дня (зміни) в залежності від його (її) тривалості, виду і категорії трудової діяльності наведений в таблиці 4.1.

Встановлено такі види трудової діяльності:

– група А – робота з зчитування інформації з екрану ПК з попереднім запитом;

- група Б – робота з введення інформації;
- група В – творча робота в режимі діалогу з ПК.

Таблиця 4.1

Час регламентованих перерв протягом робочого дня (зміни) в залежності від його (її) тривалості, виду і категорії трудової діяльності

Категорія роботи з ПК	Рівень навантаження за робочу зміну при видах роботи з ПК	Сумарний час регламентованих перерв, хвилин			
		Група В, годин	При 8-годинній зміні	При 12-годинній зміні	Сума
Група А, кількість знаків	Група Б, кількість знаків				
I	До 20 000	До 15 000	До 2,0	30	70
II	До 40 000	До 30 000	До 4,0	50	90
III	До 60 000	До 40 000	До 6,0	70	120

При виконанні протягом робочого дня робіт, що належать до різних груп, за основну роботу з ПК слід приймати таку, яка займає не менше 50% часу протягом робочого дня (зміни).

На підставі вивченої літератури по організації робочого місця користувача за комп'ютером в рамках розглянутого розділу, присвяченому охороні праці на робочому місці для користувачів персонального комп'ютера були зазначені оптимальні розміри робочого простору, а саме розміри робочого столу і крісла, робочої поверхні. Створення сприятливих умов роботи користувача за комп'ютером дозволить мінімізувати шкідливий вплив комп'ютера на здоров'я і значно збільшить продуктивність.

ВИСНОВКИ

В кваліфікаційній роботі розроблено програму для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.

В роботі були вирішені наступні задачі.

1) Проаналізовано існуючі методи для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.

2) Зроблено постановку задачі на розробку програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.

3) Розроблено ескізний проект програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.

4) Розроблено технічний проект програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.

5) Розроблено робочий проект програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Hayden J. 80% of the web powered by PHP [Electronic resource] / James Hayden. Access mode: <https://haydenjames.io/80-percent-web-powered-by-php/>
2. Boehm B. W. Software engineering economics [Text] / B. W. Boehm. – Englewood Cliffs, NJ : Prentice Hall, 1981. – 768 p. – ISBN 0-13-822122-7.
3. Oligny S. Exploring the relation between effort and duration in software engineering projects [Text] / S. Oligny, P. Bourque, A. Abran, B. Fournier // In proc. of the World Computer Congress, Aug. 2000. – P. 175-178.
4. Приходько С.Б. Розробка нелінійної регресійної моделі тривалості програмних проєктів на основі нормалізуючого перетворення Джонсона [Текст] / С. Б. Приходько, А. В. Пухалевич // Радіоелектронні і комп'ютерні системи. – Харків, 2012. – № 4 (56) – С. 90-93.
5. Prykhodko S.B. Developing PC Software Project Duration Model based on Johnson transformation [Text] / S. B. Prykhodko, A. V. Pukhalevich // Proceedings of the 12th International Conference Modern Problems of Radio Engineering, Telecommunications and Computer Science TCSET'2014, Lviv-Slavske, Ukraine. – Lviv: Polytechnic National University, 2014. – P. 114-116.
6. Tan H.B.K. Estimating LOC for information systems from their conceptual data models [Text] / H. B. K. Tan, Y. Zhao, H. Zhang // Proceedings of the 28th International Conference on Software Engineering (ICSE '06), Shanghai, China, May 20-28, 2006. – P. 321-330.
7. Tan H.B.K. Conceptual data model-based software size estimation for information systems [Text] / H. B. K. Tan, Y. Zhao, H. Zhang // Transactions on Software Engineering and Methodology. – 2009. – Vol. 19. – Issue 2. – October 2009. – Article No. 4.
8. Prykhodko N.V. Constructing the non-linear regression models on the basis of multivariate normalizing transformations [Text] / N. V. Prykhodko, S. B. Prykhodko // Electronic modeling. – 2018. – T.40. – № 6. – С. 99-108. – ISSN 0204-3572. – DOI: <https://doi.org/10.15407/emodel.40.06.101>

9. Bates D.M. Nonlinear Regression Analysis and Its Applications [Text] / D. M. Bates, D. G. Watts. – New York: John Wiley & Sons, 1988. – 384 p.
10. Seber G.A.F. Nonlinear Regression [Text] / G.A.F. Seber, C.J. Wild. – New York: John Wiley & Sons, 1989. – 768 p.
11. Ryan T.P. Modern regression methods [Text] / T. P. Ryan. – New York: John Wiley & Sons, 1997. – 529 p.
12. Drapper N.R. Applied Regression Analysis [Text] / N. R. Drapper, H. Smith. – New York: John Wiley & Sons, 1998. – 736 p.
13. Johnson R.A. Applied Multivariate Statistical Analysis [Text] / R. A. Johnson, D. W. Wichern. – Pearson Prentice Hall, 2007. – 800 p.
14. Chatterjee S. Handbook of Regression Analysis [Text] / S. Chatterjee, J. S. Simonoff. – New York: John Wiley & Sons, 2013. – 252 p.
15. Prykhodko S.B. Developing the software defect prediction models using regression analysis based on normalizing transformations [Text] / S. B. Prykhodko // Modern Problems in Testing of the Applied Software : the Research and Practice Seminar (PTASS-2016), Poltava, Ukraine, May 25-26, 2016 : abstracts. – P. 6-7.
16. Prykhodko S.B. Building the equations, confidence and prediction intervals of non-linear regressions on the basis of multivariate normalizing transformations [Text] / S. B. Prykhodko, N. V. Prykhodko, K. S. Pugachenko // Прикладні науково-технічні дослідження: матеріали II міжнар. наук.-практ. конф., 3-5 квіт. 2018 р. – Академія технічних наук України. – Івано-Франківськ: Симфонія форте, 2018. – С. 16. – ISBN 978-966-284-132-9
17. Prykhodko S. Building the Non-Linear Regression Equations on the Basis of Multivariate Normalizing Transformations [Text] / Sergiy Prykhodko, Natalia Prykhodko, Lidiia Makarova // Proceedings of the 2018 IEEE First International Conference on SYSTEM ANALYSIS & INTELLIGENT COMPUTING (SAIC), 08-12 October, 2018, Kyiv, Ukraine. – p. 48-52. – ISBN: 978-1-5386-7195-5. DOI: <https://doi.org/10.1109/SAIC.2018.8516742>
18. Prykhodko S. Estimating the Software Size of Open-Source PHP-Based Systems Using Non-Linear Regression Analysis [Text] / Sergiy Prykhodko,

Natalia Prykhodko, Lidiia Makarova // Proceedings of International Conference “Advanced Computer Information Technologies” (ACIT-2018). CEUR Workshop Proceedings. – 2019. – Vol.2300. – Ceske Budejovice, CZECH REPUBLIC. CEUR-WS.org – P. 199-202.

19. Приходько С.Б. Побудова рівняння нелінійної регресії для оцінювання розміру програмного забезпечення РНР-систем з відкритим кодом [Текст] / С. Б. Приходько, О. В. Спінов // Прикладні науково-технічні дослідження: матеріали II міжнар. наук.-практ. конф., 3-5 квіт. 2018 р. – Академія технічних наук України. – Івано-Франківськ: Симфонія форте, 2018. – С. 36. – ISBN 978-966-284-132-9

20. Спінов О.В. Оцінювання розміру програмного забезпечення РНР-систем з відкритим кодом за рівнянням нелінійної регресії [Текст] / О. В. Спінов, С. Б. Приходько // Інформатика, інформаційні системи та технології: тези доповідей п'ятнадцятої всеукраїнської конференції студентів і молодих науковців. Одеса, 27 квітня 2018 р. – Одеса, 2018. – С. 25-26.

21. Prykhodko S.B. Constructing the non-linear regression equation to estimate the software size of open source PHP-based information systems [Text] / S. B. Prykhodko, N. V. Prykhodko, T. G. Smykodub, A. V. Spinov // Проблемы информационных технологий. – 2018. – № 1 (023). – С. 118-125. – ISSN 1998-7005

22. Приходько С.Б. Нелінійна регресійна модель для оцінювання розміру Web-застосунків, що створюються з використанням фреймворку Laravel [Текст] / С. Б. Приходько, Н. В. Приходько, М. В. Ворона, І. О. Беловол // Інформаційні технології та комп'ютерна інженерія. – 2021. – Т.50. – № 1. – С. 115-121. – ISSN 1999-9941. DOI: <https://doi.org/10.31649/1999-9941-2021-50-1-115-121>

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

24. Chatterjee S. Regression Analysis by Example [Text] / S. Chatterjee, B. Price. – New York: John Wiley & Son, 1977. – 228 p.
25. Prykhodko S. Detecting Outliers in Multivariate Non-Gaussian Data on the basis of Normalizing Transformations [Text] / 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.
26. Prykhodko S. Application of the Squared Mahalanobis Distance for Detecting Outliers in Multivariate Non-Gaussian Data [Text] / 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.
27. Методичні вказівки та завдання до виконання лабораторних робіт з дисципліни "Обробка експериментальних даних на комп'ютері" [Текст] / С. Б. Приходько, Л. М. Макарова, К. С. Пугаченко. – Миколаїв: НУК, 2018. – 76 с.
28. Foss, T. A simulation study of the model evaluation criterion MMRE [Text] / T. Foss, E. Stensrud, B. Kitchenham, I. Myrtveit // IEEE Transactions on software engineering, 11(29), 2003. – P. 985–995.
29. Port, D. Comparative studies of the model evaluation criterions MMRE and PRED in software cost estimation research [Text] / D. Port, M. Korte // In: Proceedings of the 2nd ACM-IEEE International Symposium on Empirical Software Engineering and Measurement. – ACM, New York, 2008. – P. 51–60.
30. Приходько С.Б. Метод покращення нелінійних регресійних моделей на основі багатовимірних нормалізуючих перетворень [Текст] / С. Б. Приходько, Н. В. Приходько // Прикладні науково-технічні дослідження: матеріали III міжнар. наук.-практ. конф. (Івано-Франківськ, 3–5 квітня 2019 р.). – Івано-Франківськ: Сімфонія Форте, 2019. – С. 20.

31. Campbell, S.L. Modeling and Simulation in Scilab/Scicos [Text] / Stephen L. Campbell, Jean-Philippe Chancelier, Ramine Nikoukhah. – Springer, 2005. – 313 p.

32. Алексеев, Е.Р. Scilab: Решение инженерных и математических задач [Текст] / Е. Р. Алексеев, О. В. Чеснокова, Е. А. Рудченко. – М.: ALT Linux; БИНОМ. Лаборатория знаний, 2008. – 269 с.

ДОДАТОК А – Технічне завдання на розробку програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel

Вступ

Назва розробки – “Програма для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel”, далі програма.

Підстави для розробки

Завдання на випускню роботу.

Призначення розробки:

Програма призначена для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel за нелінійною регресійною моделлю.

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 Web-застосунків (значення випадкових величин): перша метрика Y включає фактичний розмір програмного забезпечення в тисячах строк коду (KLOC), друга X_1 , третя X_2 та четверта X_3 метрики визначають відповідно загальну кількість класів, середню кількість методів та суму середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти.

2) Вводити з інтерфейсу програми довірчу ймовірність, три метрики діаграми класів, за якими потрібно оцінити розмір Web-застосунку в тисячах строк коду: загальну кількість класів, середню кількість методів та суму

середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти.

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

4) Знаходити значення коефіцієнтів лінійної регресійної моделі за нормалізованими емпіричними даними.

5) Здійснювати побудову нелінійної регресійної моделі.

6) Обчислювати значення коефіцієнту детермінації R^2 .

7) Здійснювати побудову довірчого інтервалу нелінійної регресії.

8) Виводити на екран нелінійну регресію, довірчий інтервал нелінійної регресії розміру Web-застосунку в тисячах строк коду, значення коефіцієнту детермінації R^2 .

9) Виводити у текстовий файл результати оцінювання розміру Web-застосунку в тисячах строк коду.

Вимоги до організації вхідних та вихідних даних

Вхідні дані:

- емпіричні дані з метрик Web-застосунків (значення випадкових величин): перша метрика Y включає фактичний розмір програмного забезпечення в тисячах строк коду (KLOC), друга X_1 , третя X_2 та четверта X_3 метрики визначають відповідно загальну кількість класів, середню кількість методів та суму середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти;

- значення довірчої ймовірності, трьох метрик діаграми класів, за якими потрібно оцінити розмір Web-застосунку в тисячах строк коду: загальну кількість класів, середню кількість методів та суму середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти.

Вихідні дані:

- значення нелінійної регресії та границь довірчого інтервалу і

інтервалу передбачення нелінійної регресії кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.

- значення коефіцієнту детермінації.

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 маркування та пакування.

Визначаються вимогами до пакування CD дисків.

Вимоги до транспортування та зберігання.

Програма може зберігатись та транспортуватись на CD дисках.

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	27.03.2022
2. Ескізний проект	Розробка ескізного проекту	27.04.2022
	Затвердження ескізного проекту	27.04.2022
3. Технічний проект	Розробка технічного проекту	16.05.2022
	Затвердження технічного проекту	16.05.2022
4. Робочий проект	Розробка програми	23.05.2022
	Розробка програмної документації	25.05.2022
	Випробування програми	27.05.2022

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 кількості строк коду Web-застосунків,що створюються за допомогою
фреймворку Laravel
f=figure();//Создание графического окна.
//Устанавливаем размеры окна.
set(f,'position',[0,0,700,300])
//Устанавливаем заголовок окна.
set(f,'figure_name',...
'Оцінювання кількості строк коду Web-застосунків, що створюються за допомогою
фреймворку Laravel');

//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\Web_Laravel_30.txt','position',[320,240,140,20]);

BtDatInp=uicontrol('style','pushbutton','string','Ввести',...
'CallBack','[x,yx,f1,f2,f3,n]=DataInput()','position',[340,210,100,20]);

//lab_text1=uicontrol(f,'style','text','string',...
//'Параметри нормалізуючого перетворення Джонсона','position',...
//[100,270,300,20]);
lab_text2=uicontrol(f,'style','text','string',...
'Наявність фактичного розміру ПЗ','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',...
'Загальна кількість класів','position',...
[20,240,200,20]);
edit_text4=uicontrol(f,'style','edit','string',...
'171','position',[230,240,60,20]);
lab_text5=uicontrol(f,'style','text','string',...
'Середня кількість методів на клас','position',...
[20,210,200,20]);
edit_text5=uicontrol(f,'style','edit','string',...
'9.053','position',[230,210,60,20]);
lab_text6=uicontrol(f,'style','text','string',...
'Сума ср.кіл.класів,на які впливає даний клас, і ср.кіл.класів,з яких даний
клас отримує ефекти','position',...
[20,180,200,20]);
edit_text6=uicontrol(f,'style','edit','string',...
'2.485','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',...
'Результати оцінювання','position',...
[50,120,120,20]);
lab_text9=uicontrol(f,'style','text','string',...
'Результат нел. пересіи розміру ПЗ (KLOC)','position',...
[20,90,245,20]);
lab_text10=uicontrol(f,'style','text','string',...
'Нижня границя довір.інтервалу розміру (KLOC)','position',...

```

```

[20,60,245,20]);
lab_text11=uicontrol(f,'style','text','string',...
'Верхня границя довір.інтервалу розміру (KLOC)','position',...
[20,30,245,20]);
lab_text12=uicontrol(f,'style','text','string',...
'Нижня границя інтервалу прогноз. розміру (KLOC)','position',...
[355,60,260,20]);
lab_text13=uicontrol(f,'style','text','string',...
'Верхня границя інтервалу прогноз. розміру (KLOC)','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',...
'е (включити), немає (виключити)', 'value',1, 'position',...
[490,240,200,20]);
//
//radio_bikv_y=uicontrol('style','radiobutton','string',...
// 'SU (включити), SB (виключити)', 'value',1, 'position',...
//[500,180,300,20]);
//
//Проверяем значение флажка, если флажок выключен,
//if (get(radio_bikv_x, 'value')==0) & (get(radio_bikv_y, 'value')==0)
if (get(radio_bikv_x, 'value')==0)

else
//    set(textresult, 'string',...
//    'Нелінійне рівняння регресії не знайдено, Моделювання не виконано');
edit_text12=uicontrol(f,'style','edit','string',...
'0', 'position', [500,210,60,20]);
end;

Btnr_pred_int=uicontrol('style','pushbutton','string','Оцінювання', 'CallBack'
,...
'WebSoftSizeEst(x,yx,n)', 'position', [340,150,100,20]);

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

function [x, yx, f1, f2, f3, n]=DataInput()
dir_name=get(edit_text, 'string');

file_PC=mopen(dir_name, 'r');
file_PC2=mopen('D:\Data_F\Web_Laravel_30_data2.txt', 'w+');
file_PC3=mopen('D:\Data_F\Web_Laravel_30_data3.txt', 'w+');
i=1;
//n=52;
while (feof(file_PC))<1
    y(i)=mfscanf(file_PC, '%f'); 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;

    mclose(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\t%f\t%f\n', y, x1, x2, x3);
    //end;

    for i = 1:n
        for j = 1:3 fprintf(file_PC3, "%5.3f ", x(i,j)); end;
        fprintf(file_PC3, "\n");
    end;

    mclose(file_PC2);
    mclose(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: розмір (у тис.строк коду) в залежності від загальної кількості класів');
    plot2d(x1,y,-9);

    f2=figure(); //Создание графического окна.
    //Устанавливаем размеры окна.
    set(f2, 'position', [50,50,700,300]);
    //Устанавливаем заголовок окна.
    set(f2, 'figure_name', 'Емпіричні дані: розмір (у тис.строк коду) в залежності від середньої кількості методів на клас');
    plot2d(x2,y,-9);

    f3=figure(); //Создание графического окна.
    //Устанавливаем размеры окна.
    set(f3, 'position', [50,50,700,300]);
    //Устанавливаем заголовок окна.
    set(f3, 'figure_name', 'Емпіричні дані: розмір (у тис.строк коду) в залежності від суми середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти');
    plot2d(x3,y,-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]=cdf('T',Df,P,Q);

//    Delta_z=T/n^0.5;

b0=-1.74968;
b1=1.07738;
b2=0.76106;
b3=-0.19497;

Szyzyn4=0;
for i=1:n
    Szyzyn4=Szyzyn4+...
        (b0+b1*log10(x(i,1))+b2*log10(x(i,2))+b3*log10(x(i,3))-log10(yx(i,1)))^2;
end;
Szyzyn4=Szyzyn4/(n-4)
Delta_z=T*(Szyzyn4*(1/n+result))^0.5;

zy_result=b0+b1*log10(x(1,1))+b2*log10(x(1,2))+b3*log10(x(1,3));

zyLBC=zy_result-Delta_z; zyUBC=zy_result+Delta_z;

y_result=10^zy_result;
yLBC=10^zyLBC; yUBC=10^zyUBC;

file_PC5=fopen('D:\Data_F\Web_Laravel_30_data5.txt','w+');
//for i=1:3
    fprintf(file_PC5, "%5.6f ", yUBC);
//end

```

```

fclose(file_PC5);

file_PC4=fopen('D:\Data_F\Web_Laravel_30_data4.txt','w');
for i = 1:3
    for j = 1:3 fprintf(file_PC4, "%5.6f ", Szz_1(i,j)); end;
    fprintf(file_PC4, "\n");
end;
fclose(file_PC4);

endfunction

//Функція Оцінювання кількості строк коду Web-застосунків,що створюються за
//допомогою фреймворку Laravel
function WebSoftSizeEst(x, yx, 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_text7,'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; 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(x1i)-z1A; zi(2)=log10(x2i)-z2A; zi(3)=log10(x3i)-z3A;
ziT=zi';
ziSzz_1=ziT*Szz_1;
result=ziSzz_1*zi;

Df=n-4;
alpha=1-Pconf;
P=1-alpha/2;
//P=0.975;
Q=1-P;

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

b0=-1.74968;
b1=1.07738;
b2=0.76106;
b3=-0.19497;

Szyzyn4=0; Syy=0;
for i=1:n
    Szyzyn4=Szyzyn4+...
        (b0+b1*log10(x(i,1))+b2*log10(x(i,2))+b3*log10(x(i,3))-log10(yx(i,1)))^2;

    Syy=Syy+(yx(i,1)-10^b0*x(i,1)^b1*x(i,2)^b2*x(i,3)^b3)^2;
end;
Szyzyn4=Szyzyn4/(n-4)
Delta_z=T*(Szyzyn4*(1/n+result))^0.5;
Delta_z_pr=T*(Szyzyn4*(1+1/n+result))^0.5;

r_2=1-Syy/SyA;

zy_result=b0+b1*log10(x1i)+b2*log10(x2i)+b3*log10(x3i);

zyLBC=zy_result-Delta_z; zyUBC=zy_result+Delta_z;

zyLBPr=zy_result-Delta_z_pr; zyUBPr=zy_result+Delta_z_pr;

y_result=10^zy_result;
yLBC=10^zyLBC; yUBC=10^zyUBC;
yLBPr=10^zyLBPr; yUBPr=10^zyUBPr;

file_PC5=fopen('D:\Data_F\Web_Laravel_30_data5.txt','w+');
//for i=1:3
    fprintf(file_PC5, "%5.6f ", yUBC);
//end
fclose(file_PC5);

file_PC4=fopen('D:\Data_F\Web_Laravel_30_data4.txt','w+');
for i = 1:3
    for j = 1:3 fprintf(file_PC4, "%5.6f ", Szz_1(i,j)); end;
    fprintf(file_PC4, "\n");
end;
fclose(file_PC4);

set(textresult,'string', sprintf("=%1.4f",y_result));
set(textresult2,'string', sprintf("=%1.4f",yLBC));
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 (get(radio_bikv_x,'value')==0)

else
    yfact=eval(get(edit_text12,'string'));
    MRE=abs((yfact-y_result)/yfact);
    textresult5=icontrol(f,'style','text','string','', 'position',...
        [500,120,160,20]);
    set(textresult5,'string',...
        sprintf("Відносна похибка MRE=%1.4f",MRE));
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 кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel, для системи моделювання Scilab 6.0.0. Файл програми WebSoftSizeEst.sce має розмір 12 КБ.

Функціональне призначення

Програма WebSoftSizeEst призначений для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel 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 WebSoftSizeEst.sce стандартними засобами ОС.

ДОДАТОК Г – Інструкція користувача

Загальний опис роботи з ПЗ

Після завантаження файлу WebSoftSizeEst.sce на екрані з'являється форма для відображення початкового стану програми для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel (рисунок Г.1).

Рисунок Г.1 – Форма для відображення початкового стану програми

Далі у вікні редагування «Файл з даними» графічного вікна «Оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel» (рис.Г.1) потрібно відредагувати ім'я текстового файлу з початковими емпіричними даними, що будуть використовуватися для оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.

Після натискання на командну кнопку «Ввести» з текстового файлу вводяться дані (фактичний розмір ПЗ в тисячах строк коду, загальна кількість класів, середня кількість методів на клас та сума середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти), які виводяться у окремих графічних вікнах.

Потім у графічному вікні «Оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel» (рис. Г.1) у відповідних вікнах редагування потрібно відредагувати: загальну кількість класів, середню кількість методів на клас, суму середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти, довірчу ймовірність.

Крім того у графічному вікні «Оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel» (рис. Г.1) потрібно за допомогою радіо кнопки здійснити вибір наявності фактичного розміру ПЗ та ввести його для подальшого порівняння з відповідною оцінкою кількістю строк коду.

Далі потрібно натиснути на командну кнопку «Оцінювання». Після цього з'являться результати оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel.

І на останок у графічному вікні «Оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel» 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 кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel WebSoftSizeEst.

Ціль випробувань: перевірка відповідності програми WebSoftSizeEst 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 WebSoftSizeEst. 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 кількості строк коду Web-застосунків, що створюються за допомогою фреймворк...

File Tools Edit ?

Оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel

Дані для оцінювання		Файл з даними	Наявність фактичного розміру ПЗ
Загальна кількість класів	171	Data_F\Web_Laravel_30.txt	☐ Є (включити), немає (виключити)
Середня кількість методів на клас	9.053	Ввести	0
Сума ср.кіл.класів,на які впливає даний	2.485		
Довірча ймовірність	0.95	Оцінювання	
Результати оцінювання			
Результат нел. регресії розміру ПЗ (KLOC)			
Нижня границя довір.інтервалу розміру (KLOC)		Нижня границя інтервалу прогноз. розміру (KLOC)	
Верхня границя довір.інтервалу розміру (KLOC)		Верхня границя інтервалу прогноз. розміру (KLOC)	
Закрити			

Рисунок Д.1 – Графічне вікно головного модуля програми

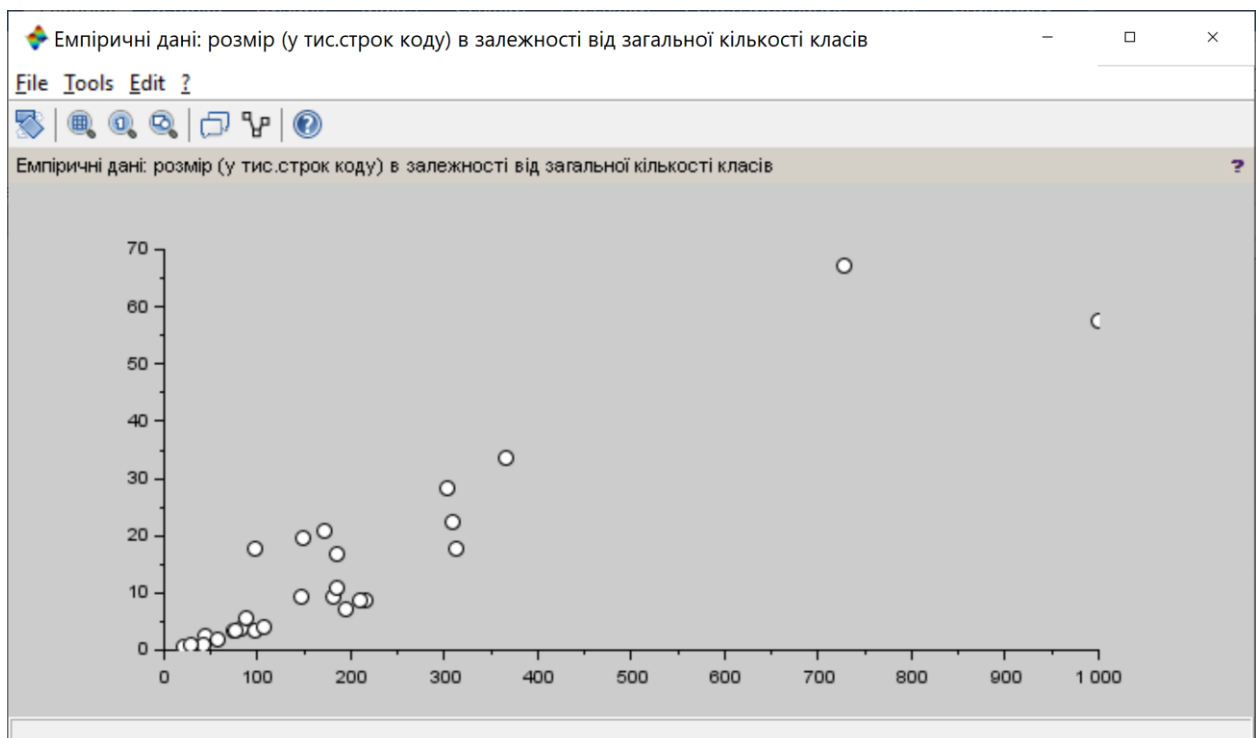


Рисунок Д.2 – Візуалізація тестових емпіричних даних.

Оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворк...

File Tools Edit ?

Оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel

Дані для оцінювання		Файл з даними	Наявність фактичного розміру ПЗ	
Загальна кількість класів	171	Data_F\Web_Laravel_30.txt	☐ є (включити), немає (виключити)	
Середня кількість методів на клас	9.053	Ввести	0	
Сума ср.кіл.класів,на які впливає дані	2.485			
Довірча ймовірність	0.95	Оцінювання		
Результати оцінювання		Коефіцієнт детермінації $R^2=0.9461$		
Результат нел. регресії розміру ПЗ (KLOC)		=20.2864		
Нижня границя довір.інтервалу розміру (KLOC)		=17.5550		Нижня границя інтервалу прогноз. розміру (KLOC) =13.8784
Верхня границя довір.інтервалу розміру (KLOC)		=23.4427		Верхня границя інтервалу прогноз. розміру (KLOC) =29.6531
Закрити				

Рисунок Д.3 – Результати роботи програми після натискання на командну кнопку «Оцінювання» у разі, коли радіо кнопка виключена

Оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворк...

File Tools Edit ?

Оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel

Дані для оцінювання		Файл з даними	Наявність фактичного розміру ПЗ
Загальна кількість класів	171	Data_F\Web_Laravel_30.txt	☒ є (включити), немає (виключити)
Середня кількість методів на клас	9.053	Ввести	20.816
Сума ср.кіл.класів,на які впливає даний	2.485		
Довірча ймовірність	0.95	Оцінювання	
Результати оцінювання		Коефіцієнт детермінації $R^2=0.9461$	
Результат нел. регресії розміру ПЗ (KLOC)		=20.2864	
Нижня границя довір.інтервалу розміру (KLOC)		=17.5550	
Верхня границя довір.інтервалу розміру (KLOC)		=23.4427	
		Нижня границя інтервалу прогноз. розміру (KLOC) =13.8784	
		Верхня границя інтервалу прогноз. розміру (KLOC) =29.6531	
Закрити			

Рисунок Д.4 – Результати роботи програми після натискання на командну кнопку «Оцінювання» у разі, коли радіо кнопка включена та введений фактичний розмір ПЗ – 20,816 тис. строк коду

Оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворк...

File Tools Edit ?

Оцінювання кількості строк коду Web-застосунків, що створюються за допомогою фреймворку Laravel

Дані для оцінювання		Файл з даними	Наявність фактичного розміру ПЗ
Загальна кількість класів	171	Data_F\Web_Laravel_30.txt	☒ є (включити), немає (виключити)
Середня кількість методів на клас	9.053	Ввести	20.816
Сума ср.кіл.класів,на які впливає даний	2.485		
Довірча ймовірність	0.99	Оцінювання	
Результати оцінювання		Коефіцієнт детермінації $R^2=0.9461$	
Результат нел. регресії розміру ПЗ (KLOC)		=20.2864	
Нижня границя довір.інтервалу розміру (KLOC)		=16.6842	
Верхня границя довір.інтервалу розміру (KLOC)		=24.6663	
		Нижня границя інтервалу прогноз. розміру (KLOC) =12.1433	
		Верхня границя інтервалу прогноз. розміру (KLOC) =33.8902	
Закрити			

Рисунок Д.5 – Результати роботи програми після натискання на командну кнопку «Оцінювання» у разі, коли радіо кнопка включена, та введене нове значення довірчої ймовірності – 0,99

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

```

20.816 171 9.053 2.485
9.482 180 3.100 4.644
57.454 1000 4.039 4.612
19.481 148 9.939 2.466
3.532 75 2.773 3.173
17.888 98 13.847 1.500
3.825 81 2.728 2.926
22.438 308 5.948 4.127
1.881 44 3.000 2.773
2.364 44 3.114 3.091
67.055 727 4.455 4.344

```

9.305 146 3.466 3.158
3.364 97 2.299 3.124
0.47 23 0.913 1.478
3.391 76 2.250 3.500
4.069 107 1.879 2.607
7.241 194 3.108 3.010
0.515 21 1.238 1.714
1.771 57 2.175 2.877
16.817 185 11.000 5.119
1.072 40 1.925 2.500
10.774 184 3.065 4.038
17.743 313 3.994 4.550
1.044 42 1.905 2.952
0.805 29 1.552 2.000
8.618 215 3.019 3.563
33.591 365 6.312 4.014
28.361 303 6.535 3.743
5.732 88 3.500 4.068
8.615 210 3.043 3.490