

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПІЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»**

**ІНСТИТУТ ЕКОНОМІКИ, УПРАВЛІННЯ ТА
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
ФАКУЛЬТЕТ ЕКОНОМІКИ І МЕНЕДЖМЕНТУ
ФОРМА НАВЧАННЯ ДЕННА
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ
ІНФОРМАТИКИ**

Допускається до захисту

Завідуючий кафедрою _____ О.О. Ємець
(підпис, ініціали, прізвище)

« ____ » _____ 2020 р.

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО БАКАЛАВРСЬКОЇ РОБОТИ**

на тему

**РОЗРОБКА ТРЕНАЖЕРУ З ТЕМИ «ЧАСТИННА КОРЕЛЯЦІЯ» ДИСЦИПЛІНИ
«АНАЛІЗ ДАНИХ І ПРИКЛАДНІ ПАКЕТИ СТАТИСТИЧНОЇ ОБРОБКИ»**

зі спеціальності 122 «Комп'ютерні науки та інформаційні технології»

Виконавець роботи

Спорнік Сергій Сергійович _____ « ____ » _____ 2020 р.
(підпис)

Науковий керівник

к.ф.-м.н, Олексійчук Юрій Федорович _____ « ____ » _____ 2020 р.
(підпис)

Полтава – 2020 р.

ЗМІСТ

| | |
|--|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ | 3 |
| ВСТУП..... | 4 |
| 1 ПОСТАНОВКА ЗАДАЧІ..... | 6 |
| 2 ІНФОРМАЦІЙНИЙ ОГЛЯД..... | 8 |
| 3 ТЕОРЕТИЧНА ЧАСТИНА..... | 10 |
| 3.1 Частинна кореляція..... | 10 |
| 3.2 Розробка алгоритму роботи тренажеру..... | 12 |
| 4 ПРАКТИЧНА РЕАЛІЗАЦІЯ..... | 19 |
| 4.1 Розробка тренажеру..... | 19 |
| 4.2 Тестування тренажеру..... | 32 |
| ВИСНОВКИ..... | 45 |
| СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ..... | 46 |
| ДОДАТОК А. КОД РОЗРОБЛЕНОГО ТРЕНАЖЕРА..... | 47 |

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

| Умовні позначення, символи, скорочення, терміни | Пояснення умовних позначень, скорочень, символів |
|--|---|
| GUI | Графічний інтерфейс користувача |
| IDE | Інтегроване середовище розробки |
| FXML | Технологія розробки графічного інтерфейсу користувача в Java |
| JavaFX | Платформа та набір інструментів для розробки графічного інтерфейсу користувача |
| Java | Крос-платформна мова програмування |
| Частинна кореляція | Ступінь зв'язку між двома випадковими величинами, коли певна множина інших випадкових величин видаляється |

ВСТУП

На кафедрі математичного моделювання та соціальної інформатики Полтавського університету економіки і торгівлі окрім денної форми навчання також є заочна та дистанційна форми. Тому використання сучасних інформаційних технологій є дуже важливим при навчанні студентів. Адже більшу частину навчального матеріалу студент повинен опрацьовувати самостійно. Це стосується як дистанційного навчання, так і заочного. Також частково це стосується і денної форми навчання – коли студент з певної причини не міг відвідувати частину занять.

Самостійне вивчення комп'ютерних та математичних дисциплін є досить складною справою. Тому на кафедрі математичного моделювання та соціальної інформатики в рамках створення дистанційних курсів розробляються навчальні тренажери [1-5]. З теми «Частинна кореляція» дисципліни «Аналіз даних і прикладні пакети статистичної обробки» тренажер до цього часу не був розроблений, хоча тема не є простою і навчальний тренажер міг би суттєво допомогти студентам. Отже, розробка тренажеру з теми «Частинна кореляція», як складової частини дистанційного навчального курсу «Аналіз даних і прикладні пакети статистичної обробки» є **актуальною**.

Метою роботи є розробка тренажеру з теми «Частинна кореляція» дисципліни «Аналіз даних і прикладні пакети статистичної обробки». Для досягнення цієї мети потрібно розв'язати такі **задачі**:

- 1) розробка алгоритму роботи тренажеру;
- 2) програмна реалізація тренажеру на деякій мові програмування;
- 3) тестування правильності роботи тренажеру.

Об'єктом дослідження є програмне забезпечення для самостійного навчання.

Предмет дослідження – програмне забезпечення тренажеру з теми «Частинна кореляція» дисципліни «Аналіз даних і прикладні пакети статистичної обробки».

Методи дослідження: методи аналізу даних, зокрема методи кореляційного аналізу, методи розробки на мові програмування Java.

Робота складається із вступу, чотирьох розділів, висновків та списку літератури. Також робота має додаток, в якому розміщений програмний код тренажеру.

В першому розділі розглядається постановка задачі та вимоги до навчального тренажеру.

В другому розділі зроблено огляд тренажерів та обґрунтовано вибір мови програмування.

В третьому розділі зроблено огляд теоретичного матеріалу, який потрібний для розробки тренажеру, розроблено алгоритм роботи тренажеру, побудована блок-схема алгоритму.

Четвертий розділ присвячений розробці тренажеру та тестуванню його роботи.

Для розробки тренажеру використовується мова програмування Java, технологія розробки графічного інтерфейсу користувача JavaFX, інтегроване середовище розробки (IDE) NetBeans.

Новизною роботи є розроблений тренажер з теми «Частинна кореляція» дисципліни «Аналіз даних і прикладні пакети статистичної обробки». Його основними перевагами є:

- ✓ простота у використанні, зрозумілий інтерфейс;
- ✓ крос-платформність;
- ✓ можливість запуску тренажеру як із серверу, так і локально на комп'ютері користувача (при цьому доступ до мережі Internet не обов'язковий);
- ✓ незначне навантаження на сервер (у випадку запуску із серверу).

1 ПОСТАНОВКА ЗАДАЧІ

Головною задачею бакалаврської роботи є створення навчального тренажеру з теми «Частинна кореляція» дисципліни «Аналіз даних і прикладні пакети статистичної обробки».

Тренажер буде працювати як доповнення до дистанційного курсу з дисципліни «Аналіз даних і прикладні пакети статистичної обробки». Тому з теоретичним матеріалом студент зможе познайомитися в лекціях дистанційного курсу. Задачею тренажеру є допомогти студенту розібратися з практичною частиною – навчитися обраховувати частинні коефіцієнти кореляції. Також передбачається, що студент вміє обраховувати звичайний коефіцієнт кореляції Пірсона.

Для знаходження частинного коефіцієнта кореляції потрібно виконувати математичні розрахунки. Тому передбачається, що студент зможе використовувати додаткове програмне забезпечення: наприклад, електронні таблиці (MS Excel або інші), калькулятор.

Тренажер буде використовуватися студентами, які самостійно вивчають тему «Частинна кореляція». Це можуть бути студенти дистанційної або заочної форм навчання, або студенти, які з певної причини пропустили відповідні заняття.

До тренажеру висувається ряд вимог.

1. Простий, зрозумілий інтерфейс. В рамках вивчення дисципліни «Аналіз даних і прикладні пакети статистичної обробки» може використовуватися багато різних тренажерів – для вивчення різних тем. Тому у студента може не бути достатньо часу та бажання розбиратися із складною програмою, читати довгі інструкції. Тренажер має бути максимально простим та зрозумілим.

2. Крос-платформність. Тренажер може використовуватися студентом в навчальній аудиторії, в бібліотеці, вдома або іншому місці. Тренажер має працювати незалежно від того, яку операційна система встановлена на комп'ютері. Також буде добре, якщо тренажер можна бути запустити як із сервера, так і локально із машини користувача.

3. Незначне навантаження на сервер. Тренажер не повинен створювати значного навантаження на сервер у випадку запуску із сервера. У випадку локального запуску тренажер має бути невеликим файлом (чи файлами), який користувач має завантажити із сервера та запускати у себе на комп'ютері.

Таким чином, було розглянуто основні вимоги до тренажеру.

В рамках бакалаврської роботи має бути розроблений алгоритм роботи тренажеру, реалізований алгоритм на вибраній мові програмуванні. Також розроблений тренажер має бути протестований на правильність роботи.

В майбутньому тренажер буде використовуватися при вивченні теми «Частинна кореляція» дисципліни «Аналіз даних і прикладні пакети статистичної обробки».

2 ІНФОРМАЦІЙНИЙ ОГЛЯД

Впровадження дистанційної форми навчання в Полтавському університеті економіки і торгівлі і, зокрема, на кафедрі математичного моделювання та соціальної інформатики призвело до необхідності розробки спеціального програмного забезпечення.

Останнім часом для багатьох дисциплін створені дистанційні курси за допомогою платформи дистанційного навчання Moodle. Але можливостей платформи Moodle не завжди достатньо для практичного навчання. Це стосується і фахових дисциплін спеціальностей «Комп'ютерні науки та інформаційні технології», «Комп'ютерні науки». Тому на кафедрі математичного моделювання та соціальної інформатики розроблена значна кількість тренажерів, зокрема [1-5].

З дисципліни «Аналіз даних і прикладні пакети статистичної обробки» розроблені навчальні тренажери з таких тем:

- 1) «Основні статистичні характеристики»;
- 2) «Кореляція якісних змінних»;
- 3) «Моделювання змінних»;
- 4) «Класифікація змінних»;
- 5) «Кореляція»;
- 6) «Парна лінійна регресія».

Всі ці тренажери створені, як окремі програми, що запускаються незалежно одна від одної.

Створення тренажерів для дисциплін, пов'язаних з математичними розрахунками, має ряд труднощів. Це пов'язано з тим, що розв'язання таких задач часто має кілька допустимих правильних шляхів. Передбачити всі ці шляхи може бути важко. Крім того, помилки в обчисленнях можуть спричинити ще більше помилок. Тому для вирішення цих проблем найчастіше використовують такий підхід: розв'язання задачі розбивають на етапи (кроки). На кожному етапі користувач отримує відносно просту задачу. Перехід до наступного етапу допускається лише після правильної відповіді. В деяких тренажерах відповідь

вибирається із запропонованих варіантів, у деяких – вводиться (в основному це стосується числових відповідей). У разі помилки користувач отримує підказку та нову спробу. Після проходження всіх кроків користувач отримує розв'язок початкової задачі.

Тренажер – це відносно невелика програма, тому для його створення можна використовувати різні мови програмування.

Особливістю тренажеру є те, що користувачі можуть запускати його на своєму комп'ютері. Тому тренажер має бути крос-платформним, щоб не створювати проблем користувачам при його запуску.

В якості мови програмування вибрана мова програмування Java.

Мова програмування Java є [12-15]:

1. Об'єктно-орієнтованою: в Java все є об'єктами. Програма може бути легко розширена, оскільки вона базується на об'єктній моделі.

2. Крос-платформна: на відміну від багатьох інших мов (C, C++, Delphi) Java не компілюється в машинний код конкретної операційної системи, а в незалежний від платформи байт-код. Цей байт-код інтерпретується за допомогою віртуальної машини Java (Java Virtual Machine – JVM). На сьогоднішній день JVM існують для всіх основних операційних систем.

3. Простою – не дивлячись на потужні можливості, Java є відносно простою у вивченні та використанні.

4. Багатопотоковою – можна створювати програми, які будуть виконуватися одночасно в кількох потоках.

5. Високопродуктивною: використання Just-In-Time (JIT) компілятора підвищує швидкість роботи програм, написаних на Java.

6. Популярною – Java одна із найбільш популярних мов програмування. Тому існує велика кількість бібліотек, підручників, прикладів використання різних програмних конструкцій.

Тому мова програмування Java чудово підходить для створення навчального тренажеру.

3 ТЕОРЕТИЧНА ЧАСТИНА

3.1 Частинна кореляція

Коефіцієнт кореляції

Розглянемо необхідний для розробки алгоритму роботи тренажеру теоретичний матеріал. [6-11]

Умовне математичне сподівання $M_x(Y)$ — математичне сподівання випадкової величини Y , що його обчислено у припущенні, що змінна X прийняла значення x .

Кореляційною залежністю між двома змінними називається функціональна залежність між значеннями однієї з них та умовним математичним сподіванням іншої.

Числовою характеристикою кореляційної залежності є коефіцієнт кореляції.

Коефіцієнт кореляції вперше був запропонований Карлом Пірсоном в 1896 році. Коефіцієнт кореляції Пірсона r (або просто – коефіцієнт кореляції) характеризує силу лінійного зв'язку між двома змінними. За означенням коефіцієнт кореляції теоретично може приймати будь-яке значення з відрізка $[-1, 1]$.

Значення коефіцієнту кореляції трактується так:

- 1) якщо $r=0$, то це означає відсутність лінійного зв'язку;
- 2) якщо $r=1$, то це означає наявність ідеального (функціонального) прямого лінійного зв'язку. Тобто збільшення однієї змінної призводить до збільшення іншої за деяким лінійним законом;
- 3) якщо $r=-1$, то це означає наявність ідеального (функціонального) зворотного лінійного зв'язку. Тобто збільшення однієї змінної призводить до зменшення іншої за деяким лінійним законом;
- 4) значення коефіцієнта кореляції між 0 і 0,3 (0 і -0,3) вказує на слабкий прямий (зворотний) лінійний зв'язок;
- 5) значення коефіцієнта кореляції між 0,3 і 0,7 (-0,3 і -0,7) вказує на помірний прямий (зворотний) лінійний зв'язок;
- 6) значення коефіцієнта кореляції між 0,7 і 1 (-0,7 і -1) вказує на сильний прямий (зворотний) лінійний зв'язок.

Зауваження. Використання коефіцієнт кореляції доцільне лише у випадку, коли зв'язок між змінними лінійний. Якщо зв'язок лінійний, то коефіцієнт кореляції добре описує його силу. Якщо зв'язок нелінійний, то використання коефіцієнту кореляції немає смислу.

Для обчислення коефіцієнту кореляції можна використати формулу:

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}, \quad (3.1)$$

де

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (3.2)$$

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i \quad (3.3)$$

Частинна кореляція

Часто спостерігається ситуація, коли не пов'язані між собою змінні мають велику кореляцію. Це може бути як додатна, так і від'ємна кореляція. Як правило, це пояснюється наявністю іншої змінної, яка сильно корелює із обома змінними.

Для того щоб вплив кореляційного зв'язку між двома змінними "очистити" від можливого впливу третьої, уведене поняття **частинної кореляції**.

Частинні коефіцієнти кореляції так само, як і парні, характеризують тісноту зв'язку між двома змінними. Але, на відміну від парних, частинні коефіцієнти характеризують тісноту зв'язку за умови, що інші незалежні змінні сталі.

При використанні частинного коефіцієнта кореляції необхідно пам'ятати:

- взаємовпливаючих змінних може бути не три, а скільки завгодно;
- ви можете не знати про усі взаємовпливаючі змінні.

Частинним коефіцієнтом кореляції для змінних $x^{(i)}$, $x^{(j)}$ будемо називати величину:

$$r_{ij}^* = -\frac{R_{ij}}{\sqrt{R_{ii}R_{jj}}} \quad (3.4)$$

де R_{ij} – алгебраїчне доповнення для елемента (i, j) у звичайній кореляційній матриці:

$$R = \begin{pmatrix} 1 & r_{01} & \dots & r_{0q} \\ r_0 & 1 & \dots & r_{1q} \\ \dots & \dots & \dots & \dots \\ r_{q0} & r_{q1} & \dots & 1 \end{pmatrix}$$

де r_{ij} - звичайний коефіцієнт кореляції.

Властивості частинного коефіцієнта кореляції співпадають з властивостями звичайного коефіцієнта кореляції.

3.2 Розробка алгоритму роботи тренажеру

Навчальний тренажер з теми «Частинна кореляція» призначений для студентів, які вивчають дисципліну «Аналіз даних і прикладні пакети статистичної обробки». Головна мета тренажеру – навчити студентів обраховувати частинний коефіцієнт кореляції.

Перед використанням тренажеру передбачається, що студент знайомий з теоретичним матеріалом з відповідної теми. Також студент повинен вміти обраховувати парний коефіцієнт кореляції та алгебраїчне доповнення до матриці.

Користувачу буде запропоновано набір даних для трьох змінних (x_1, x_2, x_3). Користувач повинен буде знайти частинну кореляцію для кожної із пар (x_1x_2, x_1x_3, x_2x_3). Розв'язання прикладу буде розбите на кроки, на кожному кроці буде запропоновано конкретне питання, на яке студент має дати відповідь. У випадку неправильної відповіді, виводиться підказка. У випадку правильної відповіді – відбувається перехід на наступний етап.

Розглянемо роботу тренажеру на прикладі.

Користувачу виводяться дані (табл. 3.1).

Таблиця 3.1 Початкові дані задачі

| X1 | X2 | X3 |
|----|----|-----|
| 1 | 2 | 0 |
| 5 | -3 | -4 |
| 7 | 7 | -8 |
| 9 | 9 | -12 |
| 8 | 11 | -11 |
| 10 | 10 | -15 |
| 12 | 18 | -17 |

Користувачу виводиться запитання: "Чому рівна кореляція між змінними X1 та X2 (з точністю до трьох знаків після коми)?"

Відповідь повинна бути введена у спеціальне поле. Для обрахунку коефіцієнта кореляції студент може використовувати допоміжні інструменти – наприклад, програму Excel. Якщо користувач вводить неправильну відповідь, то йому дається підказка. Якщо користувач вводить правильну відповідь (0.825), то відбувається перехід до наступного кроку.

На наступному кроці користувачу виводиться запитання: "Чому рівна кореляція між змінними X1 та X3 (з точністю до трьох знаків після коми)?"

Тут правильна відповідь: -0,986.

Якщо відповідь введена неправильна, то даються підказки аналогічно до попереднього кроку.

Після введення правильної відповіді користувач бачить запитання: "Чому рівна кореляція між змінними X2 та X3 (з точністю до трьох знаків після коми)?"

В даному випадку правильна відповідь: -0,873.

Після обрахунку цих трьох кореляцій користувач отримає часткову заповнену кореляційну матрицю (табл. 3.2).

Таблиця 3.2 Частково заповнена кореляційна матриця

| | | |
|----|-------|--------|
| | 0.825 | -0.986 |
| r= | | -0.873 |
| | | |

Після цього користувач повинен заповнити кореляційну матрицю повністю. У випадку неправильної відповіді буде виведена підказка про те, що кореляційна матриця симетрична, а по головній діагоналі повинні бути одиниці. В результаті буде отримана кореляційна матриця (табл. 3.3)

Таблиця 3.3 Кореляційна матриця

$$r = \begin{array}{|c|c|c|} \hline & 1 & 0.825 & -0.986 \\ \hline 0.825 & & 1 & -0.873 \\ \hline -0.986 & -0.873 & & 1 \\ \hline \end{array}$$

Далі користувачу потрібно знайти допоміжні матриці, які будуть використовуватися для обчислення алгебраїчних доповнень.

Користувачу по чергово будуть виводитися наступні запитання:

"Введіть елементи матриці, яка буде одержана при вилученні з матриці r рядка номер 1 та стовпця номер 1".

Правильна відповідь:

$$R_{11} = \begin{array}{|c|c|} \hline & 1 & -0.873 \\ \hline -0.873 & & 1 \\ \hline \end{array}$$

"Введіть елементи матриці, яка буде одержана при вилученні з матриці r рядка номер 2 та стовпця номер 2".

Правильна відповідь:

$$R_{22} = \begin{array}{|c|c|} \hline & 1 & -0.986 \\ \hline -0.986 & & 1 \\ \hline \end{array}$$

"Введіть елементи матриці, яка буде одержана при вилученні з матриці r рядка номер 3 та стовпця номер 3".

Правильна відповідь:

$$R_{33} = \begin{array}{|c|c|} \hline & 1 & 0.825 \\ \hline \end{array}$$

$$\begin{array}{|c|c|} \hline 0.825 & 1 \\ \hline \end{array}$$

"Введіть елементи матриці, яка буде одержана при вилученні з матриці r рядка номер 1 та стовпця номер 2".

Правильна відповідь:

$$R_{12} = \begin{array}{|c|c|} \hline 0.825 & -0.873 \\ \hline -0.986 & 1 \\ \hline \end{array}$$

"Введіть елементи матриці, яка буде одержана при вилученні з матриці r рядка номер 1 та стовпця номер 3".

Правильна відповідь:

$$R_{13} = \begin{array}{|c|c|} \hline 0.825 & 1 \\ \hline -0.986 & -0.873 \\ \hline \end{array}$$

"Введіть елементи матриці, яка буде одержана при вилученні з матриці r рядка номер 2 та стовпця номер 3".

Правильна відповідь:

$$R_{23} = \begin{array}{|c|c|} \hline 1 & 0.825 \\ \hline -0.986 & -0.873 \\ \hline \end{array}$$

У випадку неправильної відповіді на кожному кроці студенту виводиться підказка.

Після знаходження допоміжних матриць, обраховуються алгебраїчні доповнення до відповідних елементів. Алгебраїчні доповнення обраховуються, як визначники матриць, помножені на коефіцієнт $(-1)^{(i+j)}$.

Користувачу будуть виводитися наступні питання:

"Знайдіть алгебраїчне доповнення до елемента матриці r з рядка номер 1 та стовпця номер 1 (з точністю до трьох знаків після коми)"

Правильна відповідь: 0,238.

"Знайдіть алгебраїчне доповнення до елемента матриці r з рядка номер 2 та стовпця номер 2 (з точністю до трьох знаків після коми)"

Правильна відповідь: 0,027.

"Знайдіть алгебраїчне доповнення до елемента матриці r з рядка номер 3 та стовпця номер 3 (з точністю до трьох знаків після коми)"

Правильна відповідь: 0,319.

"Знайдіть алгебраїчне доповнення до елемента матриці r з рядка номер 1 та стовпця номер 2 (з точністю до трьох знаків після коми)"

Правильна відповідь: 0,035.

"Знайдіть алгебраїчне доповнення до елемента матриці r з рядка номер 1 та стовпця номер 3 (з точністю до трьох знаків після коми)"

Правильна відповідь: 0,266.

"Знайдіть алгебраїчне доповнення до елемента матриці r з рядка номер 2 та стовпця номер 3 (з точністю до трьох знаків після коми)"

Правильна відповідь: 0,059.

Як і раніше, при неправильній відповіді користувачу буде виводитися підказка.

Після того, як усі потрібні алгебраїчні доповнення будуть знайдені, можна переходити до обчислення частинних коефіцієнтів кореляції за формулою (3.4).

Користувачу будуть виводитися наступні питання:

"Обчисліть частинний коефіцієнт кореляції між змінними x_1 та x_2 (з точністю до трьох знаків після коми)".

Правильна відповідь: -0,439.

"Обчисліть частинний коефіцієнт кореляції між змінними x_1 та x_3 (з точністю до трьох знаків після коми)".

Правильна відповідь: -0,964.

"Обчисліть частинний коефіцієнт кореляції між змінними x_2 та x_3 (з точністю до трьох знаків після коми)".

Правильна відповідь: -0,630.

Після обчислення всіх частинних коефіцієнтів кореляції виводиться повідомлення про успішне проходження тренажеру.

Отже, роботу тренажеру можна описати наступним алгоритмом.

Алгоритм роботи тренажеру:

1. Виводиться стартове вікно тренажеру.

2. Виводиться набір даних.

3. Користувач обраховує по чергово парні коефіцієнти кореляції. Якщо коефіцієнт обрахований правильно, то перехід на крок 4. Інакше – виводиться повідомлення із підказкою і повернення на крок 3.

4. Користувач вводить кореляційну матрицю, яка складається із коефіцієнтів кореляції, обрахованих на попередньому кроці. Якщо матриця введена правильно, то перехід на крок 5. Інакше – виводиться повідомлення із підказкою і повернення на крок 4.

5. Користувач по чергово вводить допоміжні матриці для обчислення алгебраїчних доповнень. Якщо матриці введено правильно, то перехід на крок 6. Інакше – виводиться повідомлення із підказкою і повернення на крок 5.

6. Користувач обчислює алгебраїчні доповнення для матриці парних коефіцієнтів кореляції. Якщо алгебраїчні доповнення введено правильно, то перехід на крок 7. Інакше – виводиться повідомлення із підказкою і повернення на крок 6.

7. Користувач обчислює частинні коефіцієнти кореляції. Якщо частинні коефіцієнти кореляції введено правильно, то перехід на крок 8. Інакше – виводиться повідомлення із підказкою і повернення на крок 7.

8. Завершення роботи алгоритму.

Для алгоритму побудована блок-схема (рис. 3.1).

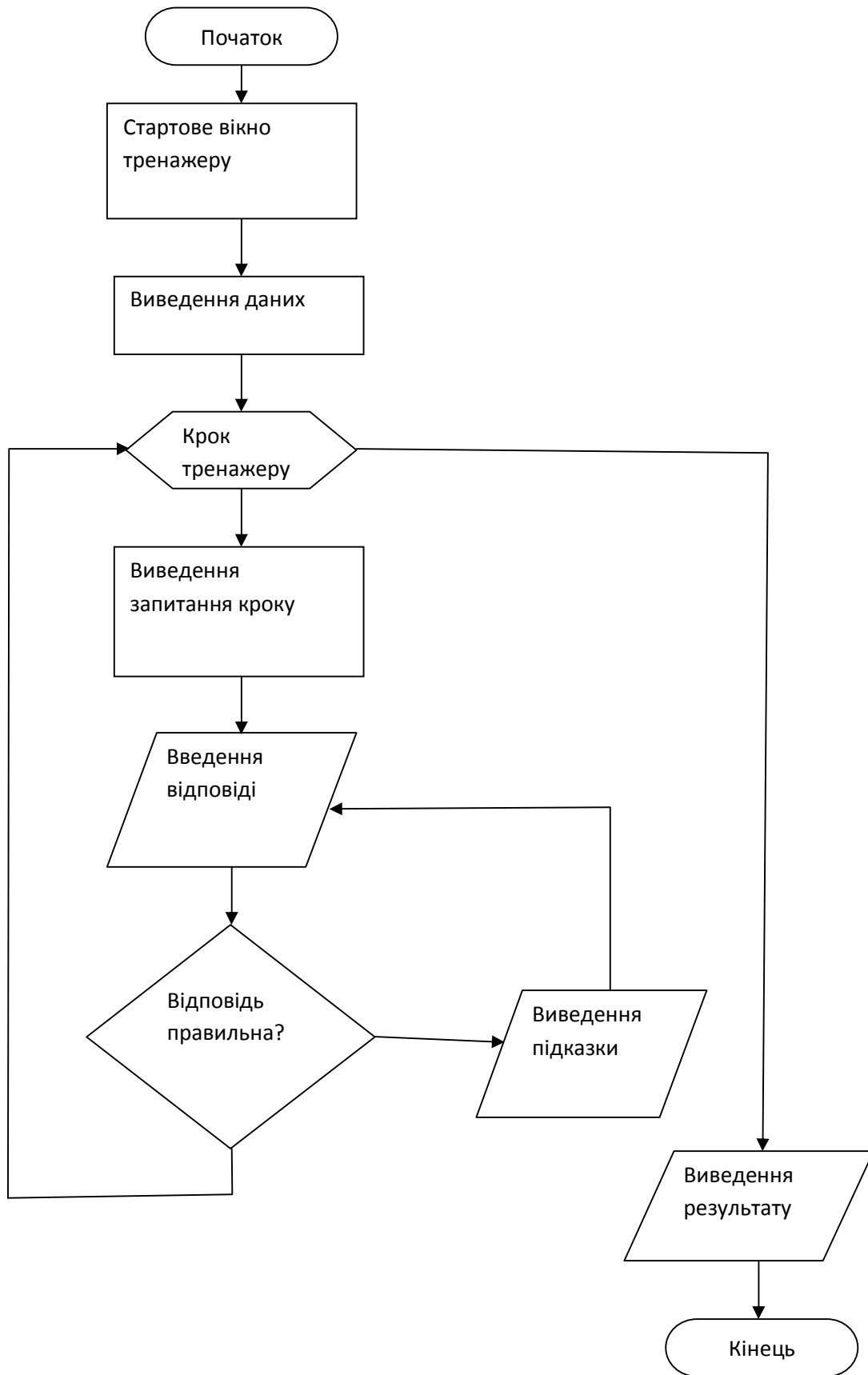


Рис 3.1 Блок-схема алгоритму роботи тренажеру

4 ПРАКТИЧНА РЕАЛІЗАЦІЯ

4.1 Розробка тренажеру

Для розробки навчального тренажеру вибрана мова програмування Java. Невід'ємною складовою тренажеру є графічний інтерфейс користувача (GUI – graphical user interface). Для розробки графічного інтерфейсу вибрана технологія JavaFX з використанням XML-подібної мови розмітки FXML.

Графічний інтерфейс тренажеру створений таким чином, щоб всі значення користувач вводив лише в одне поле в верхній частині вікна (рис. 4.1). Це вікно виділене жовтим кольором. Після введення значення користувач може натиснути кнопку "Ввести" або клавішу Enter.

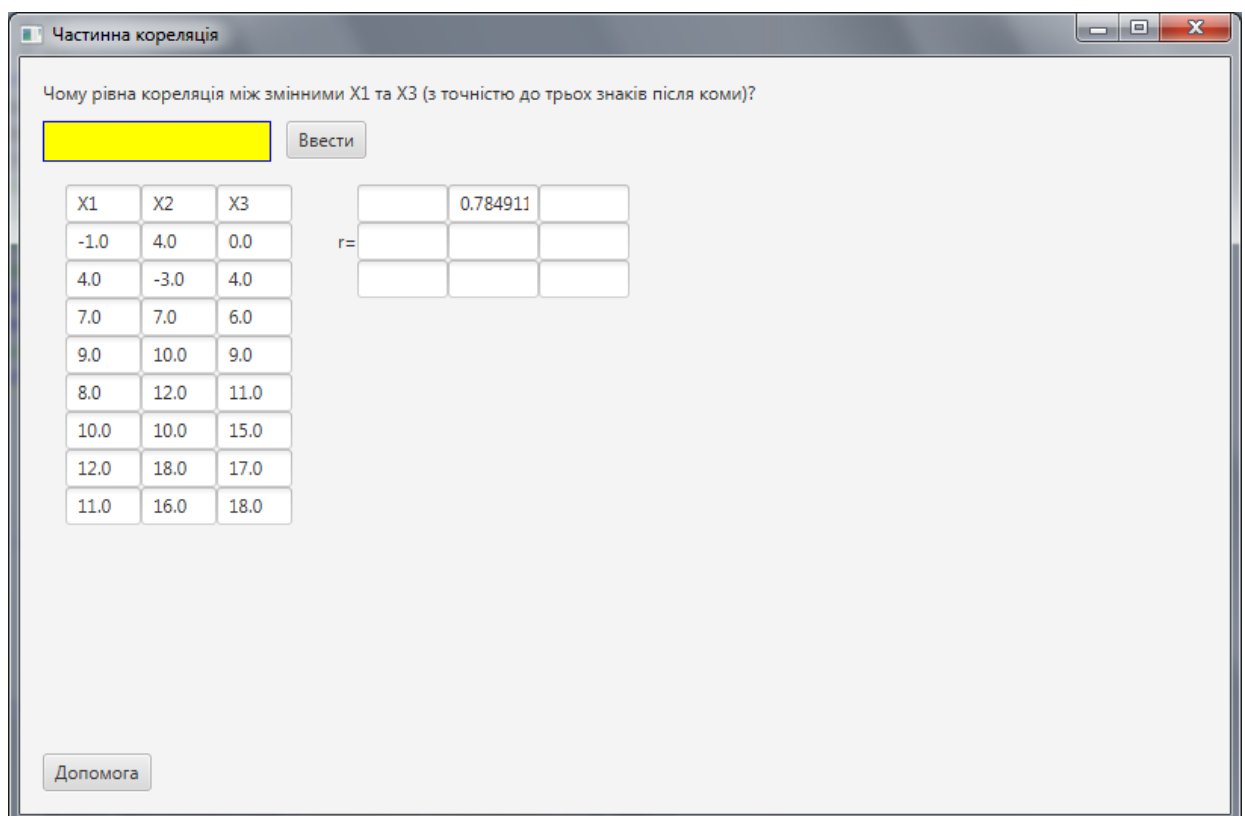


Рис. 4.1 Вигляд тренажеру

В лівій частині вікна виводиться набір даних. Всього є 3 набори даних, в кожному із них по 3 змінних, але різна кількість вимірів. За потреби можуть бути додані нові набори даних, зміни в коді програми буду незначними.

В центральній частині тренажеру поступово буде виводитися розв'язок прикладу.

В нижній частині знаходиться кнопка "Допомога". Після натиску на неї з'являється вікно із короткою інструкцією та інформацією про тренажер (рис. 4.2).

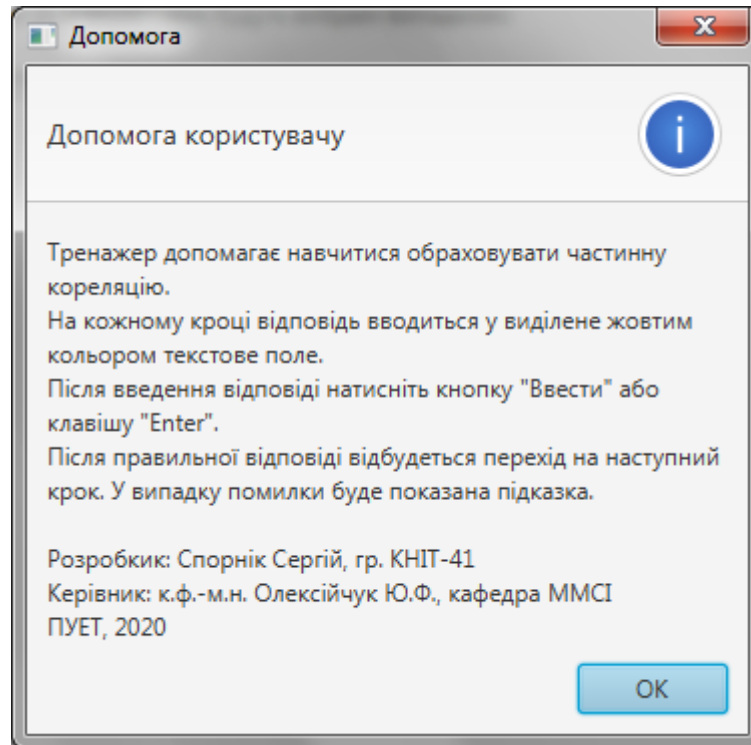


Рис. 4.1 Вікно допомоги

Важливим моментом розробки тренажеру є проблема округлення при обчисленнях. Для розв'язання задачі користувач може використовувати додаткові програми (калькулятор, Excel), але відповідь на кожному кроці вводиться з певною точністю. Тому помилки можуть накопичуватися. Для вирішення цієї проблеми використано наступний підхід:

1) відповідь користувача приймається, у випадку, якщо вона відрізняється від розрахованої тренажером не більше ніж на 0.0025;

2) якщо користувач ввів правильну відповідь, то в тренажері показується не ця відповідь, а розрахована тренажером. Для подальших розрахунків користувач може використовувати ці більш точні числа.

Проект містить 4 файли – 3 класи і FXML-файл:

- 1) FXMLDocument.fxml;
- 2) Main.java;
- 3) Data.java;
- 4) FXMLDocumentController.java.

Розглянемо їх.

1) **FXMLDocument.fxml**

Для взаємного розміщення елементів керування у FXML-файлі використано `BorderPane`.

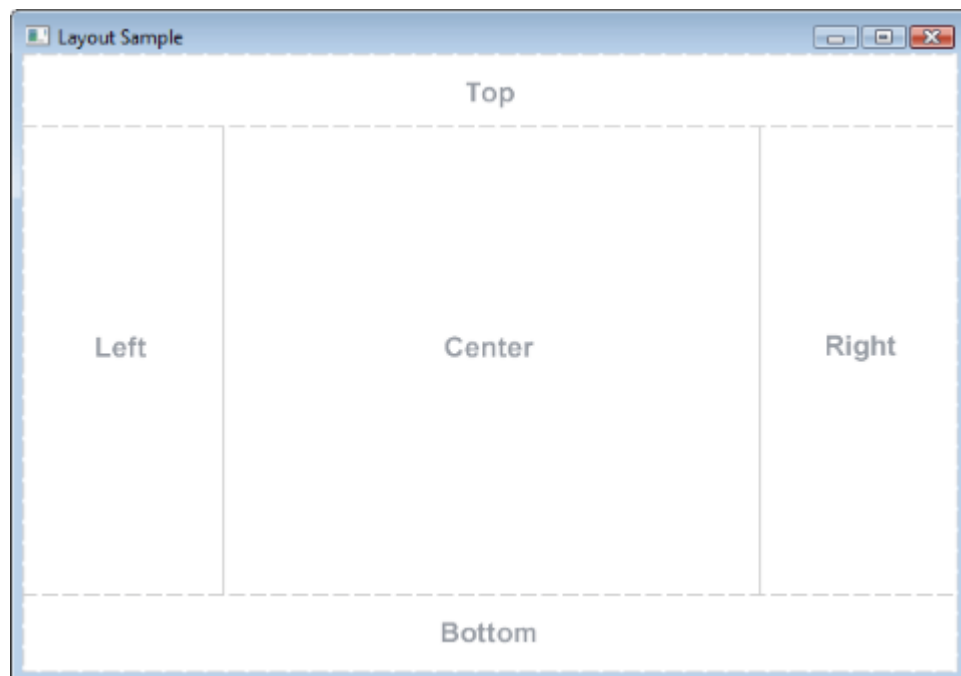


Рис. 4.1 Загальний вигляд `BorderPane`

`BorderPane` (рис 4.1) має 5 областей, де можуть розміщуватися елементи керування.

Більшість елементів будуть додаватися динамічно під час виконання програми.

У верхній частині (`top`) розміщується блок, в якому будуть виводитися питання кожного етапу, поле для ведення відповіді та кнопка для відправки відповіді. В

основному користувач буде взаємодіяти (вводити відповіді) лише із цією частиною вікна. Код блоку:

```

<top>
  <VBox spacing="10">
    <Label fx:id="info" text="Виберіть номер даних (1 або 2 або 3). Інакше
- дані будуть вибрані випадково:"/>
    <HBox spacing="10">
      <TextField fx:id="input" onAction="#nextStep"/>
      <Button fx:id="enter" text="Ввести" onAction="#nextStep"/>
    </HBox>
  </VBox>
</top>

```

В лівій частині будуть виводитися дані, для яких треба буде розрахувати частинну кореляцію. Код блоку:

```

<left>
  <VBox spacing="10">
    <padding><Insets top="15" right="15" bottom="15" left="15"/></padding>
    <GridPane fx:id="dataGrid">
    </GridPane>
  </VBox>
</left>

```

В центральній частині (center) буде поетапно виводитися розв'язок задачі. Код цього блоку займає багато місця, тому тут не показаний. Ознайомитися із ним можна в додатку.

В нижній частині розміщена кнопка "Допомога":

```
<bottom>
  <Button fx:id="help" text="Допомога" onAction="#help"/>
</bottom>
```

2) Main.java

Клас Main містить стандартний код для JavaFX-програм.

Основним тут є метод start(), який створює та запускає вікно тренажеру:

```
public void start(Stage stage) throws Exception {
    Parent root = FXMLLoader.load(getClass().getResource(
"FXMLDocument.fxml"));
    Scene scene = new Scene(root);
    stage.setScene(scene);
    stage.setTitle("Частинна кореляція");
    stage.show();
}
```

3) Data.java

В класі Data описуються набори даних та основні обрахунки, необхідні для обрахунку частинної кореляції.

Клас містить такі поля:

```
private int cols;//кількість змінних
private int rows;//кількість вимірів для кожної змінної
private String name;
private double [][] corMatrix;//кореляційна матриця
private List <double []> data=new ArrayList();
```

Також клас Data містить наступні методи:

void addVariable(double [] x) – додає нову змінну в набір даних; код методу:

```
public void addVariable(double [] x){
```

```

if(cols==0){
    data.add(x);
    cols++;
    rows=x.length;
} else {
    if(x.length==rows){
        data.add(x);
        cols++;
    }
    else{
        throw new RuntimeException("Змінна має іншу кількість значень");
    }
}
}

```

`void calculateCorMatrix()` – розраховує кореляційну матрицю; код методу:

```

public void calculateCorMatrix(){
    corMatrix=new double [cols][cols];
    for(int i=0;i<cols;i++){
        for(int j=0;j<cols;j++){
            corMatrix[i][j]=calculateCorrelation(data.get(i),data.get(j));
        }
    }
}

```

`double[][] getCorMatrix()` – повертає кореляційну матрицю (перед цим матриця обраховується); код методу:

```

public double[][] getCorMatrix(){

```



```

    calculateCorMatrix();
    return corMatrix;
}

```

double calculateCorrelation(double [] a, double [] b) – обраховує коефіцієнт кореляції для двох масивів даних; код методу:

```

public double calculateCorrelation(double [] a, double [] b){
    if(a.length!=b.length){
        throw new RuntimeException("Масиви мають різну кількість елементів");
    }
    double sumA=0;
    for(double d:a){
        sumA+=d;
    }
    double sumB=0;
    for(double d:b){
        sumB+=d;
    }
    double meanA=sumA/a.length;
    double meanB=sumB/b.length;
    double sA2=0;
    double sB2=0;
    double sAB=0;
    for(int i=0; i<a.length;i++){
        sA2+=(a[i]-meanA)*(a[i]-meanA);
        sB2+=(b[i]-meanB)*(b[i]-meanB);
        sAB+=(a[i]-meanA)*(b[i]-meanB);
    }
}

```

```

return sAB/Math.sqrt(sA2*sB2);
}

```

double [][] calculateMatrixR(int i, int j) – обраховує допоміжну матрицю для обрахунку алгебраїчного доповнення; код методу:

```

public double [][] calculateMatrixR(int i, int j){
    double [][]r=new double [corMatrix.length-1][corMatrix.length-1];
    int rr=0;
    int cc=0;
    for(int row=0;row<corMatrix.length;row++){
        for(int col=0;col<corMatrix.length;col++){
            if(row!=(i-1)&&col!=(j-1)){
                r[row-rr][col-cc]=corMatrix[row][col];
            }else{
                if(row==i-1){
                    rr=1;
                }
                if(col==j-1){
                    cc=1;
                }
            }
        }
        cc=0;
    }
    return r;
}

```

double calculateCofactor(int i, int j) – обраховує алгебраїчне доповнення до елемента матриці; код методу:

```

public double calculateCofactor(int i, int j){
    double [][]r=calculateMatrixR(i, j);
    return Math.pow(-1, i+j)*(r[0][0]*r[1][1]-r[0][1]*r[1][0]);
}

```

double calculatePartCor(int i, int j) – обраховує частинну кореляцію; код методу:

```

public double calculatePartCor(int i, int j){
    double cor = -calculateCofactor(i, j) / Math.sqrt(calculateCofactor(i,i) *
calculateCofactor(j,j));
    return cor;
}

```

int getCol() – повертає кількість змінних в наборі даних; код методу:

```

public int getCol() {
    return cols;
}

```

int getRow() – повертає кількість рядків у даних; код методу:

```

public int getRow() {
    return rows+1;
}

```

String getValue(int i, int j) – повертає відповідний елемент із даних; код методу:

```

public String getValue(int i, int j){
    if(j==0){

```

```

        return name+(i+1);
    }else{
        return data.get(i)[j-1]+"";
    }
}

```

`double getCorrelation(int i,int j)` – повертає елемент із кореляційної матриці. Код методу:

```

public double getCorrelation(int i,int j){
    return corMatrix[i][j];
}

```

4) FXMLDocumentController.java

Клас FXMLDocumentController зв'язує графічний інтерфейс користувача та логіку роботи програми. Цей клас можна вважати основним. Також цей клас має найбільший обсяг. Розглянемо лише деякі його елементи.

Дані, для яких треба обрахувати частинні кореляції, задаються в методі `initialize()`, який запускається при завантаженні класу. Наприклад:

```

data1 = new Data("X");
double[] x11 = {1, 5, 7, 9, 8, 10, 12};
double[] x12 = {2, -3, 7, 9, 11, 10, 18};
double[] x13 = {0, -4, -8, -12, -11, -15, -17};
data1.addVariable(x11);
data1.addVariable(x12);
data1.addVariable(x13);

```

Наступний метод виводить дані користувачу:

```

private void showData(GridPane dataGrid, Data X) {
    dataGrid.getChildren().clear();
}

```

```

TextField tf;
for (int i = 0; i < X.getCol(); i++) {
    for (int j = 0; j < X.getRow(); j++) {
        tf = new TextField();
        dataGrid.add(tf, i, j);
        tf.setMaxWidth(50);
        tf.setText(String.valueOf(X.getValue(i, j)));
        tf.setEditable(false);
    }
}
}

```

Метод, який виводить шаблон для матриці, яку буде поступово буде заповнювати користувач:

```

private void showMatrix(GridPane grid, double[][] matrix) {
    grid.getChildren().clear();
    TextField tf;
    for (int i = 0; i < matrix.length; i++) {
        for (int j = 0; j < matrix[i].length; j++) {
            tf = new TextField();
            grid.add(tf, j, i);
            tf.setMaxWidth(60);
            tf.setEditable(false);
        }
    }
}
}

```

Коли користувач надсилає відповідь на питання, то відбувається виклик методу `nextStep()`. Цей метод виконує певні дії в залежності від номеру кроку (всього 40 різних кроків). Наприклад, для кроку 3 код має наступний вигляд.

Спочатку перевіряється номер кроку:

```
if (step == 3) {
```

Потім зчитується введене користувачем значення:

```
    try {  
        inputDouble = Double.parseDouble(input.getText());
```

Якщо значення не можна перетворити в тип `double`, то виводиться повідомлення про некоректність введених даних:

```
    } catch (Exception e) {  
        Alert alert = new Alert(Alert.AlertType.ERROR);  
        alert.setTitle("Некоректне введення");  
        alert.setHeaderText("Некоректне введення");  
        alert.setContentText("Потрібно ввести число.");  
        alert.showAndWait();  
        return;  
    }
```

Далі обраховується правильна відповідь на поточному кроці:

```
    int i = 1;  
    int j = 2;  
    double cor = data.getCorrelation(i, j);
```

Правильна відповідь порівнюється з введеною користувачем. Якщо модуль різниці не перевищує допустиму похибку, правильне значення виводиться у потрібне місце на формі:

```
if (Math.abs(inputDouble - cor) < eps) {
    TextField tf = (TextField) corMatrix.getChildren().get(i * 3 + j);
    tf.setText(cor + "");
```

Збільшується номер кроку на 1 то кількість допущених помилок на поточному кроці стає рівною 0:

```
step++;
errors=0;
```

Виводиться питання наступного кроку і метод завершує свою роботу:

```
info.setText("Чому рівна кореляція між змінними X2 та X1 (з  
точністю до трьох знаків після коми)?");
input.setText("");
return;
```

Якщо модуль різниці перевищує допустиму похибку, то кількість допущених помилок збільшуємо на 1:

```
} else {
    errors++;
```

В залежності від кількості допущених помилок на поточному кроці, виводиться повідомлення з підказками. Якщо користувач допустив більше двох помилок, то йому показується правильна відповідь:

```

if (errors == 1) {
    Alert alert = new Alert(Alert.AlertType.ERROR);
    alert.setTitle("Неправильна відповідь");
    alert.setHeaderText("Неправильна відповідь");
    alert.setContentText("На жаль, неправильно. Спробуйте ще");
    alert.showAndWait();
} else if (errors == 2) {
    Alert alert = new Alert(Alert.AlertType.ERROR);
    alert.setTitle("Неправильна відповідь");
    alert.setHeaderText("Неправильна відповідь");
    alert.setContentText("Потрібно порахувати кореляцію. Кореляція
може бути в межах від -1 до 1.");
    alert.showAndWait();
} else {
    Alert alert = new Alert(Alert.AlertType.ERROR);
    alert.setTitle("Неправильна відповідь");
    alert.setHeaderText("Неправильна відповідь");
    alert.setContentText("Правильна відповідь: " + cor);
    alert.showAndWait();
}
}
}

```

Повний код проекту знаходиться в додатку.

4.2 Тестування тренажеру

Важливим етапом розробки програмного забезпечення є тестування. Під час тестування перевіряється правильність роботи програмного забезпечення.

На рис. 4.3 показано стартове вікно тренажеру.

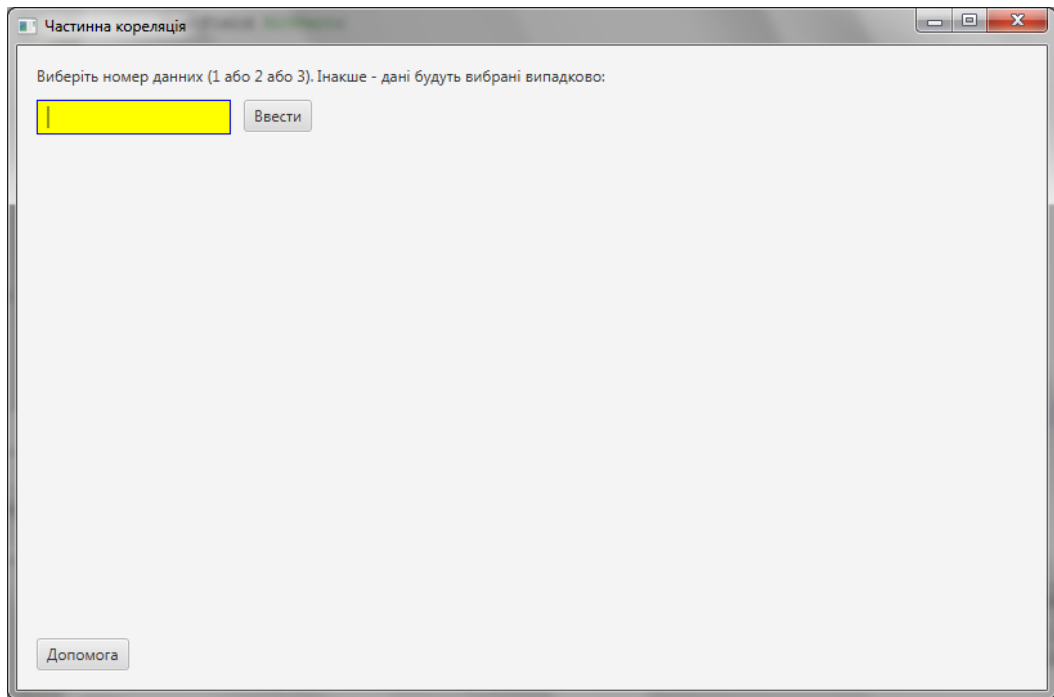


Рис. 4.3 Стартове вікно тренажеру

Користувач може ввести цифри 1, 2 або 3, щоб отримати один із трьох наборів даних. У випадку якщо користувач просто натисне кнопку "Ввести" (або введе щось інше), то набір даних вибирається випадковим чином. Перевіряємо різні варіанти, отримуємо результати (рис. 4.4 - 4.7).

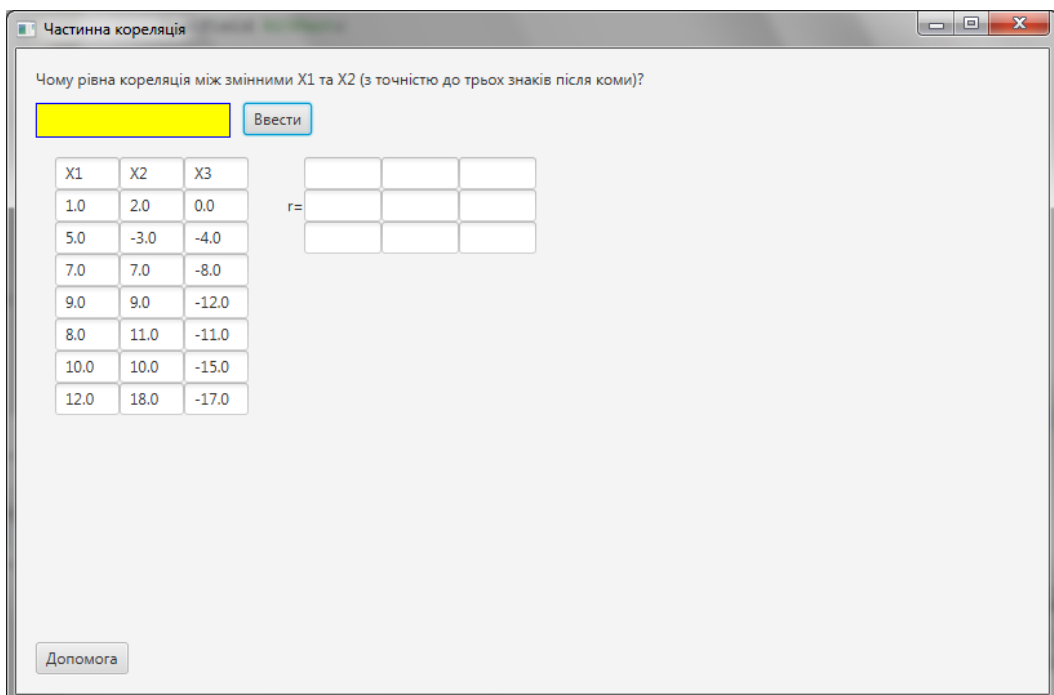


Рис. 4.4 Набір даних №1

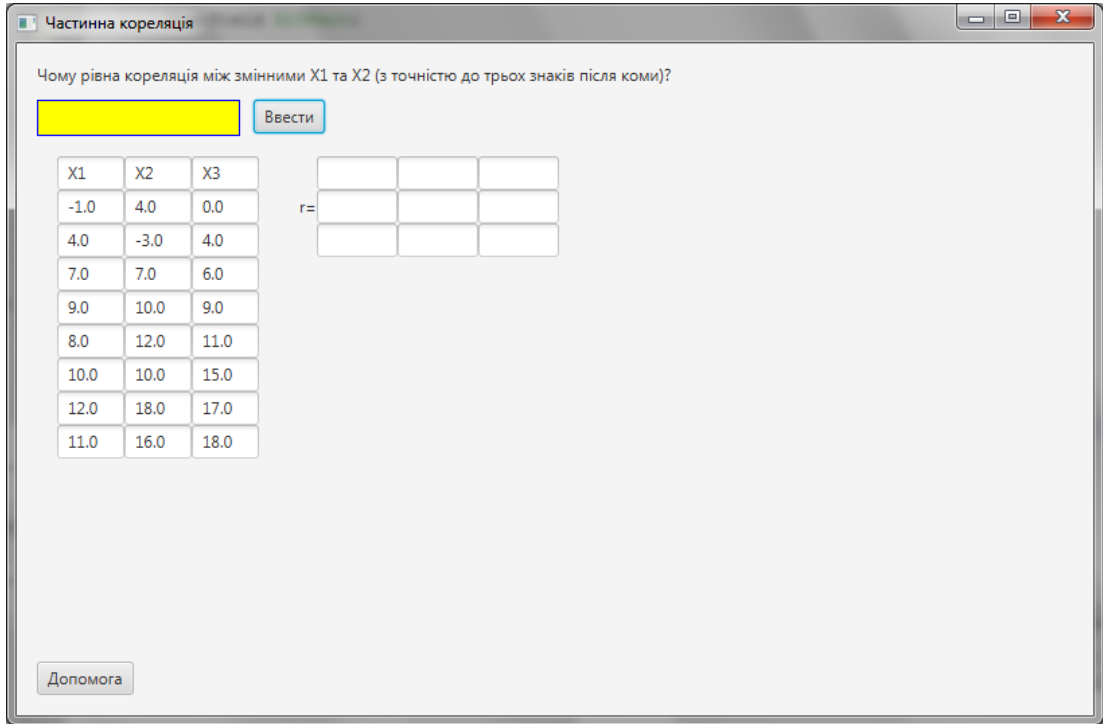


Рис. 4.5 Набір даних №2

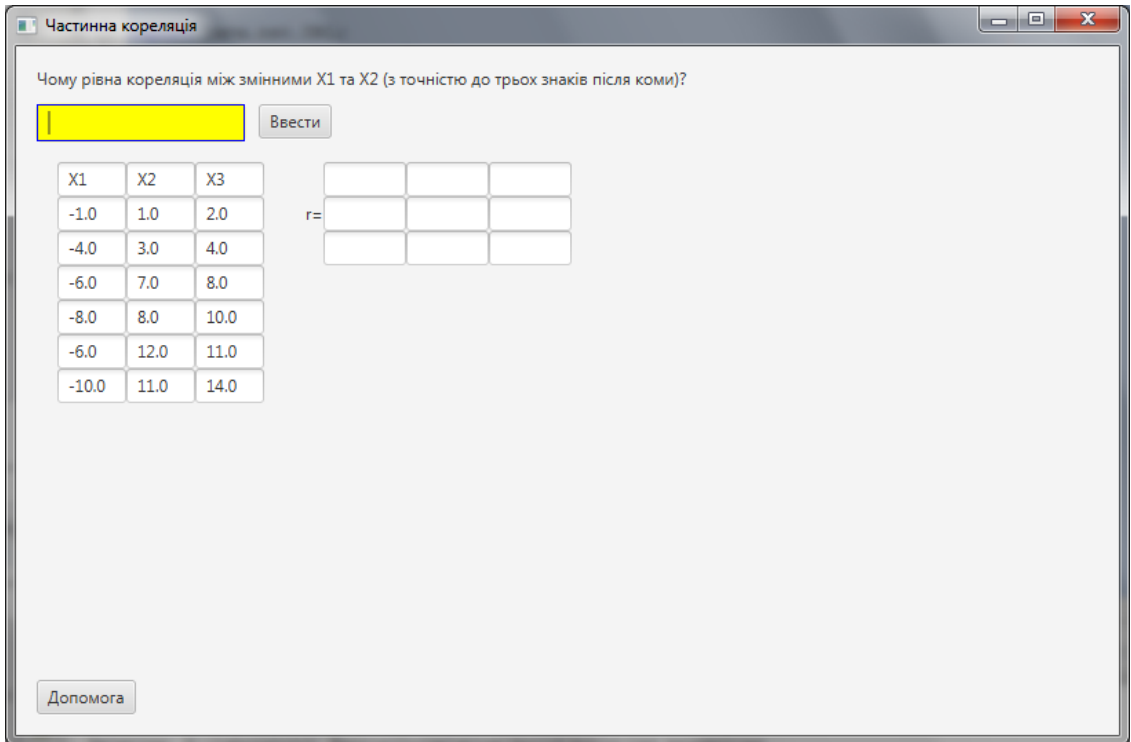


Рис. 4.6 Набір даних №3

Частинна кореляція

Чому рівна кореляція між змінними X1 та X2 (з точністю до трьох знаків після коми)?

Ввести

| X1 | X2 | X3 |
|-------|------|------|
| -1.0 | 1.0 | 2.0 |
| -4.0 | 3.0 | 4.0 |
| -6.0 | 7.0 | 8.0 |
| -8.0 | 8.0 | 10.0 |
| -6.0 | 12.0 | 11.0 |
| -10.0 | 11.0 | 14.0 |

г=

Допомога

Рис. 4.7 Випадково вибраний набір даних

Також при випадковому виборі перевірено, що кожного разу набір даних дійсно вибирався випадково.

Частинна кореляція

Чому рівна кореляція між змінними X1 та X2 (з точністю до трьох знаків після коми)?

Ввести

| X1 | X2 | X3 |
|------|------|-------|
| 1.0 | 2.0 | 0.0 |
| 5.0 | -3.0 | -4.0 |
| 7.0 | 7.0 | -8.0 |
| 9.0 | 9.0 | -12.0 |
| 8.0 | 11.0 | -11.0 |
| 10.0 | 10.0 | -15.0 |
| 12.0 | 18.0 | -17.0 |

г=

Допомога

Рис. 4.8 Перший крок тренажера

На наступному етапі потрібно порахувати кореляцію між змінними X_1 та X_2 (рис. 4.8). У випадку помилкової відповіді буде виведено вікно з інформацією про помилку (рис. 4.9). Після повторної помилки з'являється інша інформація (рис. 4.10). Після третьої – показується правильна відповідь (рис. 4.11).

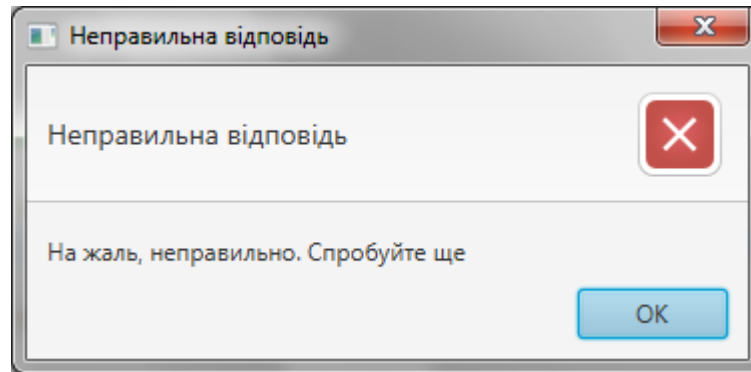


Рис. 4.9 Вікно з інформацією про помилку

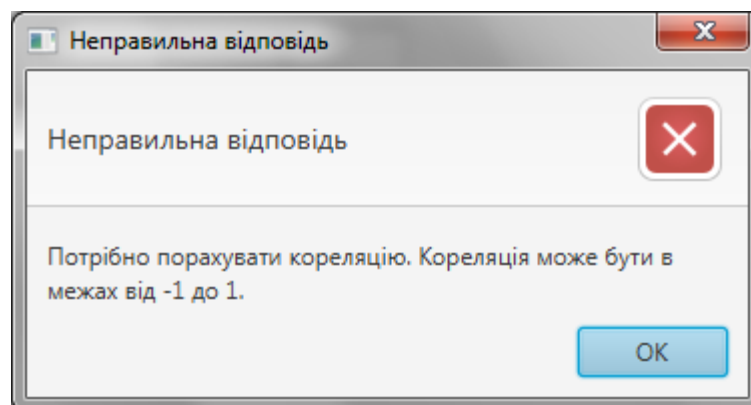


Рис. 4.10 Інформація після другої помилки

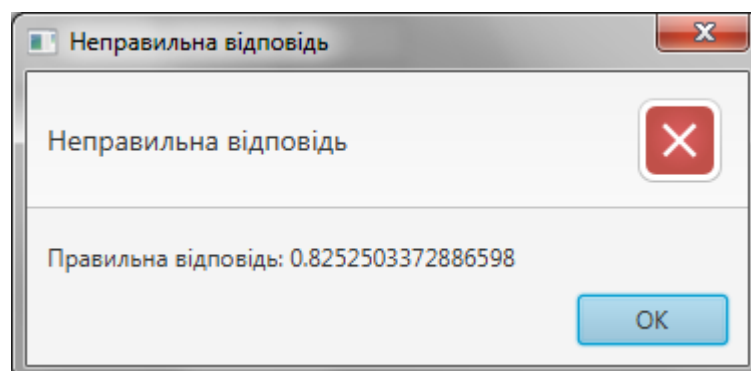


Рис. 4.11 Інформація після третьої помилки

Аналогічно обчислюються інші парні кореляції. У випадку обрахунку кореляцій для пар змінних, для яких вони вже були обраховані, виводяться інші підказки. Наприклад, коли користувач повинен обрахувати кореляцію між X_2 та X_1 (кореляція між X_1 та X_2 вже обчислена), то у випадку другої помилки виводиться інше повідомлення (рис. 4.12), а після третьої помилки правильна відповідь не показується, а виводиться підказка, як на рис. 4.13.

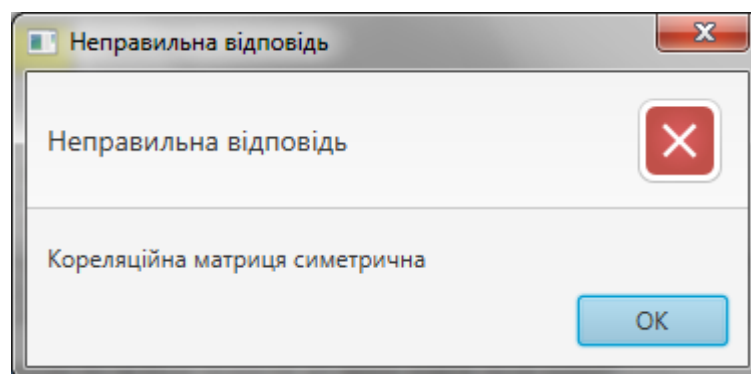


Рис. 4.12 Інформація після другої помилки

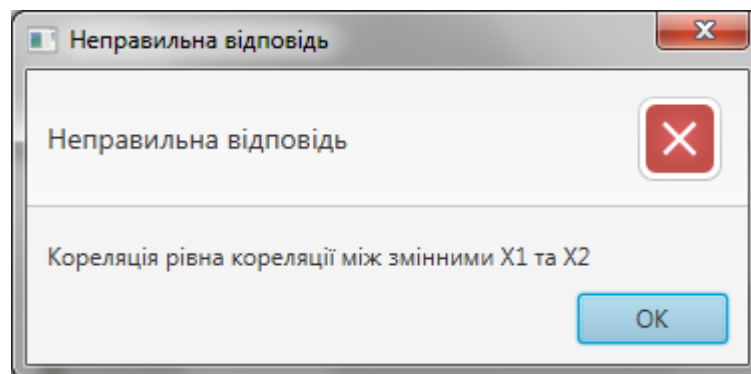


Рис. 4.13 Інформація після третьої помилки

На наступному етапі потрібно ввести діагональні елементи кореляційної матриці (рис. 4.14). Оскільки це питання просте, то всі діагональні елементи заповнюються за один крок.

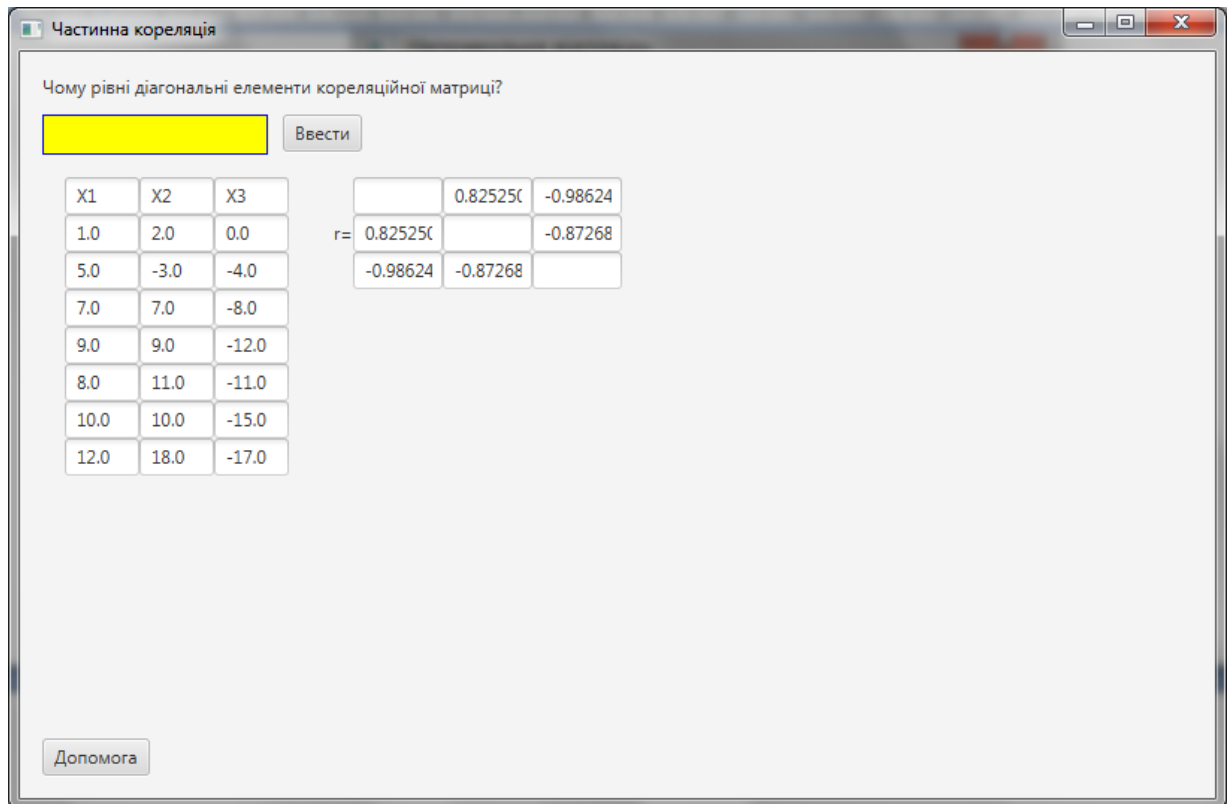


Рис. 4.14 Обрахунок діагональних елементів кореляційної матриці

Відмітимо, що у випадку введення нечислового виразу або числа в некоректній формі, виводиться відповідне інформаційне повідомлення (рис. 4.15)

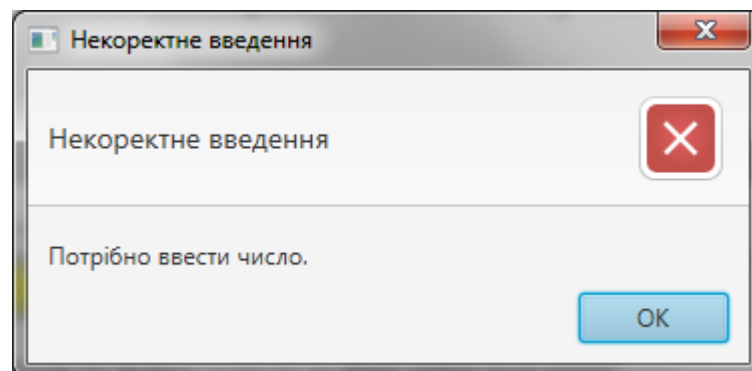


Рис. 4.15 Інформація при некоректному введенні

Після введення діагональних елементів користувачу потрібно поетапно заповнювати елементи матриці R11 (рис. 4.16).

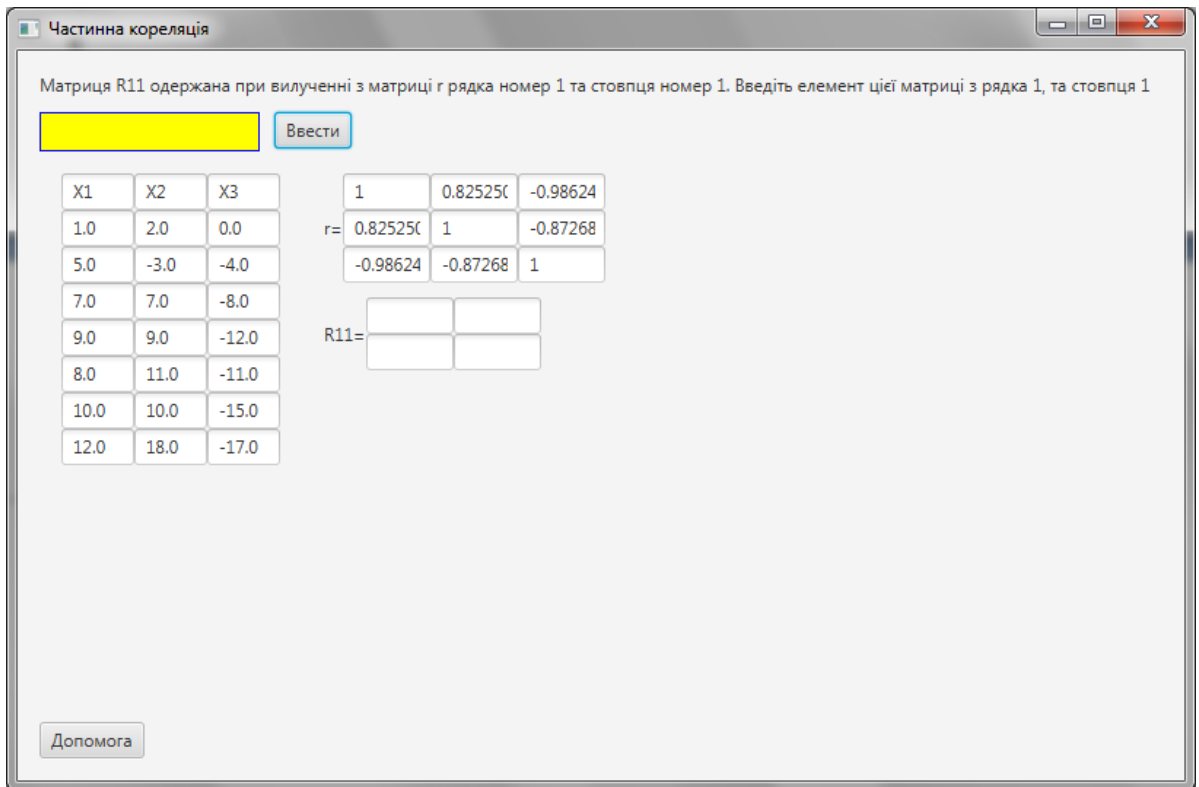


Рис. 4.16 Вигляд тренажеру перед заповненням елементів матриці R11

У випадку введення користувачем першої неправильної відповіді виводиться стандартне вікно про помилку (рис. 4.9). У випадку другої помилки – вікно, як на рис. 4.17, після третьої помилки – рис. 4.18.

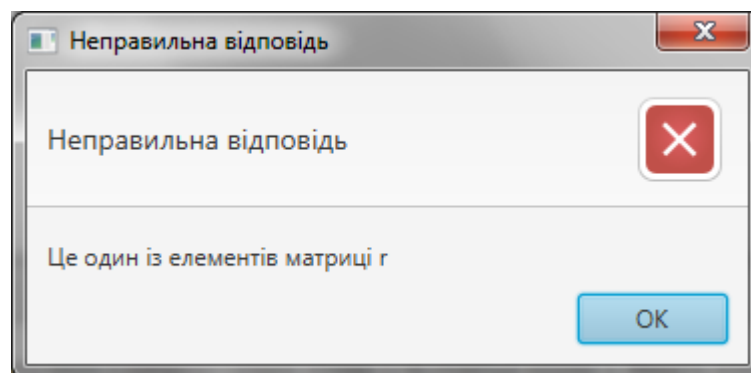


Рис. 4.17 Інформація після другої помилки при заповненні R11

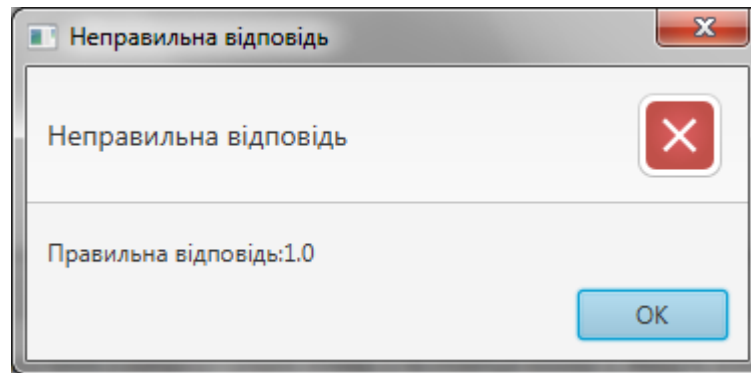


Рис. 4.18 Інформація після третьої помилки при заповненні R11

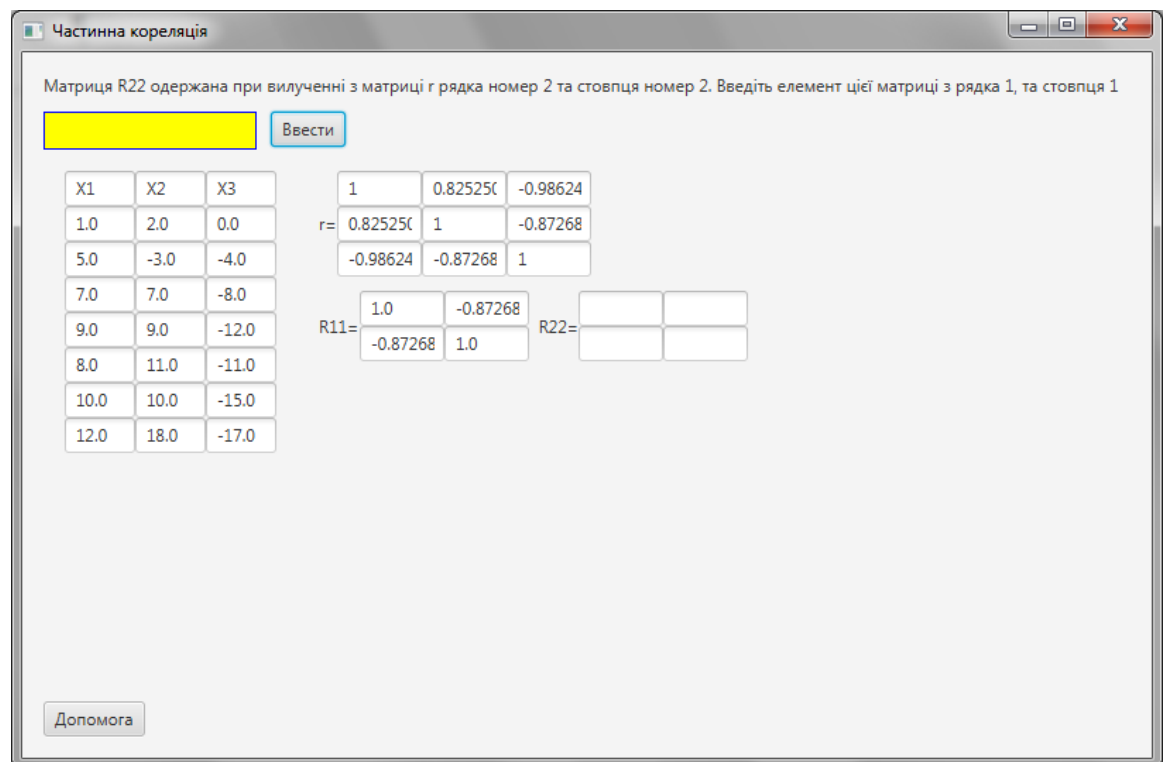


Рис. 4.19 Вигляд тренажеру після заповнення матриці R11

Аналогічно заповнюються елементи матриць R22, R33, R12, R13, R23. Відмітимо, що користувач може просто копіювати відповідні значення із матриці r , щоб прискорити введення відповіді.

Наступний етап – обрахунок алгебраїчних доповнень. Використовуючи матриці знайдені на попередньому етапі, користувач повинен обрахувати відповідні алгебраїчні доповнення.

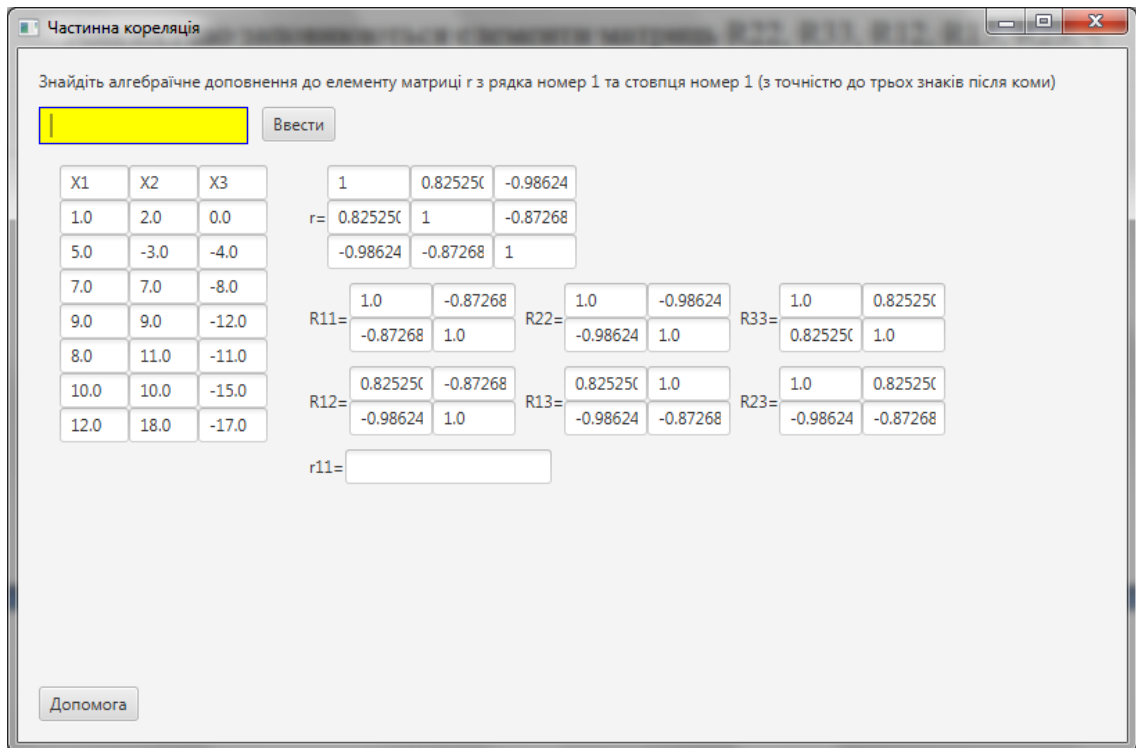


Рис. 4.20 Вигляд тренажеру перед знаходженням алгебраїчних доповнень

Після першої неправильної відповіді користувач отримує стандартне повідомлення (рис. 4.9). Після другої та третьої помилок – повідомлення як на рис. 4.21 та 4.22 відповідно.

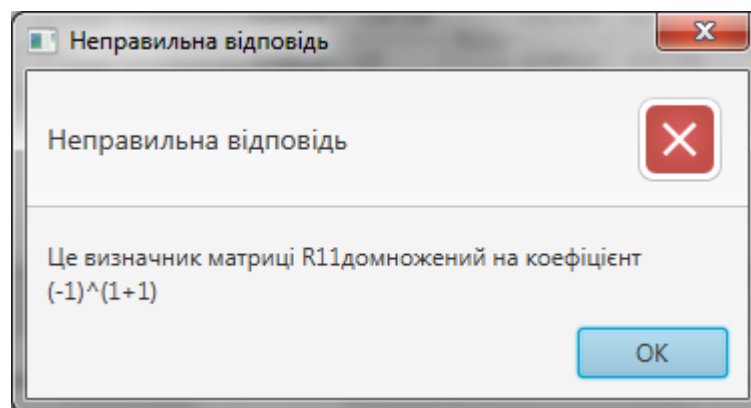


Рис. 4.21 Вікно з підказкою після другої помилки при обрахунку алгебраїчного доповнення

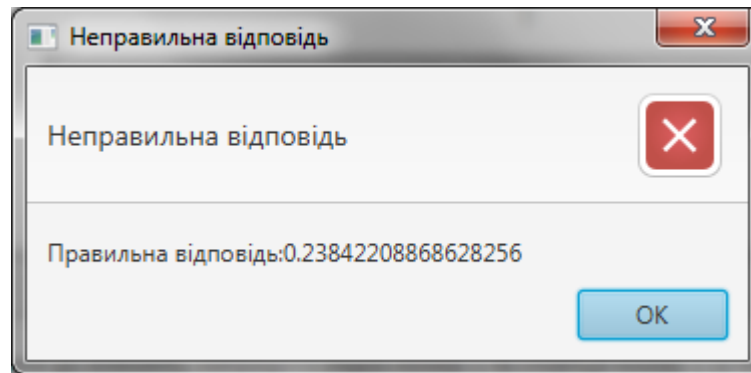


Рис. 4.22 Вікно з підказкою після третьої помилки при обрахунку алгебраїчного доповнення

Аналогічно обраховуються алгебраїчні доповнення до інших елементів кореляційної матриці (рис. 4.23).

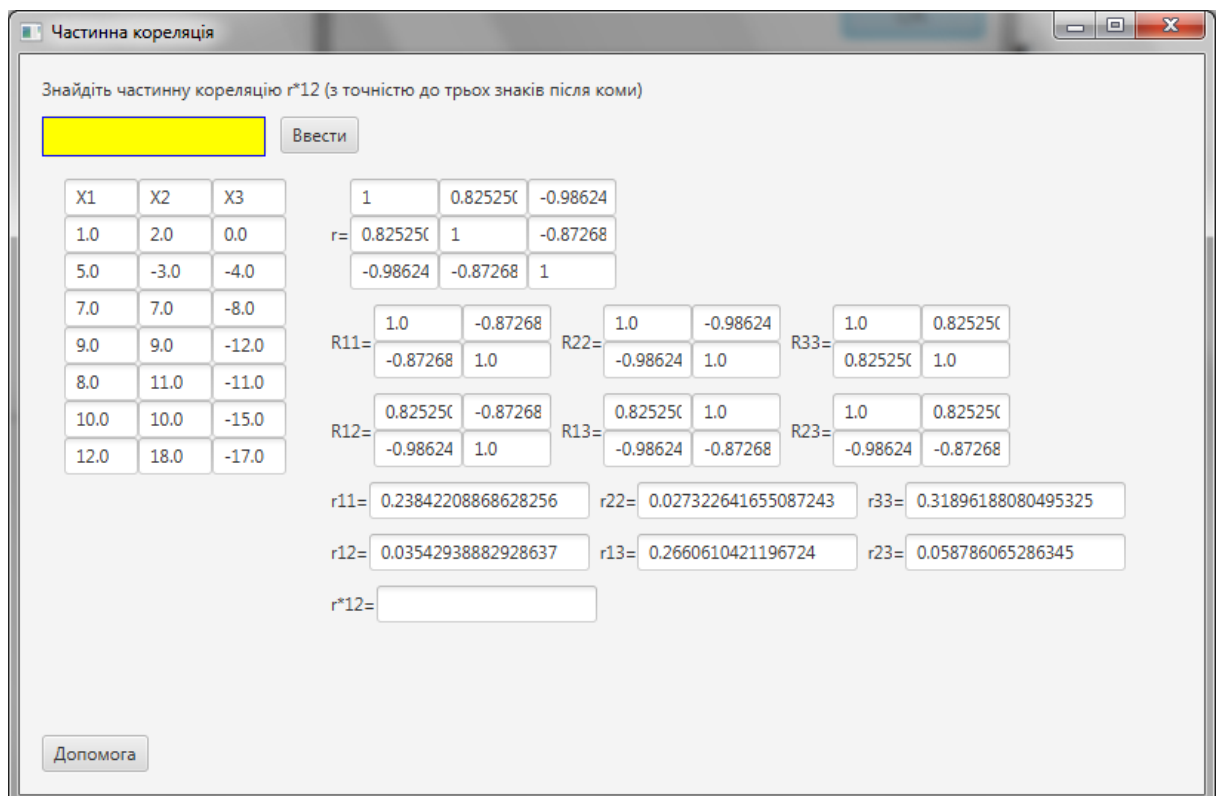


Рис. 4.23 Вікно тренажеру після обрахунку алгебраїчних доповнень

Після цього користувач переходить до обрахунку частинних кореляцій (рис. 4.23). У випадку введення неправильної відповіді виводиться стандартне вікно з інформацією про помилку (рис. 4.9). Якщо знову введена неправильна відповідь, то

виводиться підказка із формулою для обрахунку частинної кореляції (рис. 4.24). Якщо наступна введена відповідь знову буде неправильна, то буде виведене повідомлення із правильною відповіддю (рис. 4.25).

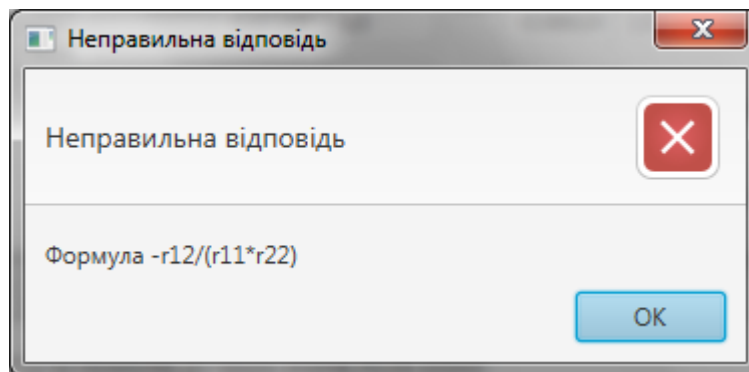


Рис. 4.24 Підказка із формулою для обрахунку частинної кореляції

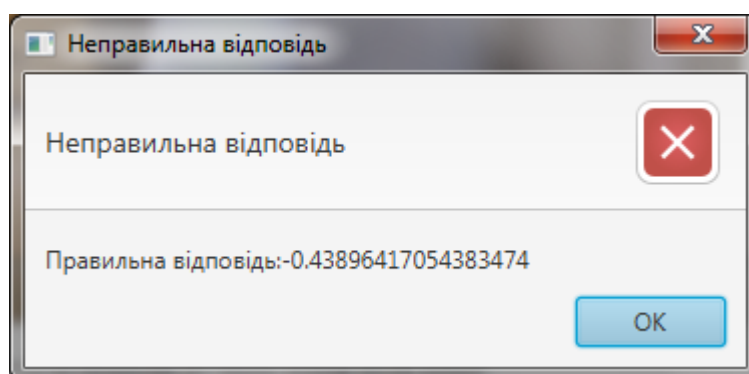


Рис. 4.25 Підказка із правильною відповіддю

Аналогічно обраховуються інші частинні кореляції. Після цього проходження тренажеру завершується (рис. 4.26).

Якщо користувач бажає знову пройти тренажер з початку, то він має закрити вікно і запустити тренажер знову.

Тренажер розроблений таким чином, що користувач може вводити дані лише в одне поле. Інші поля недоступні для введення, хоча з них можна скопіювати значення. Це суттєво спрощує тестування тренажеру.

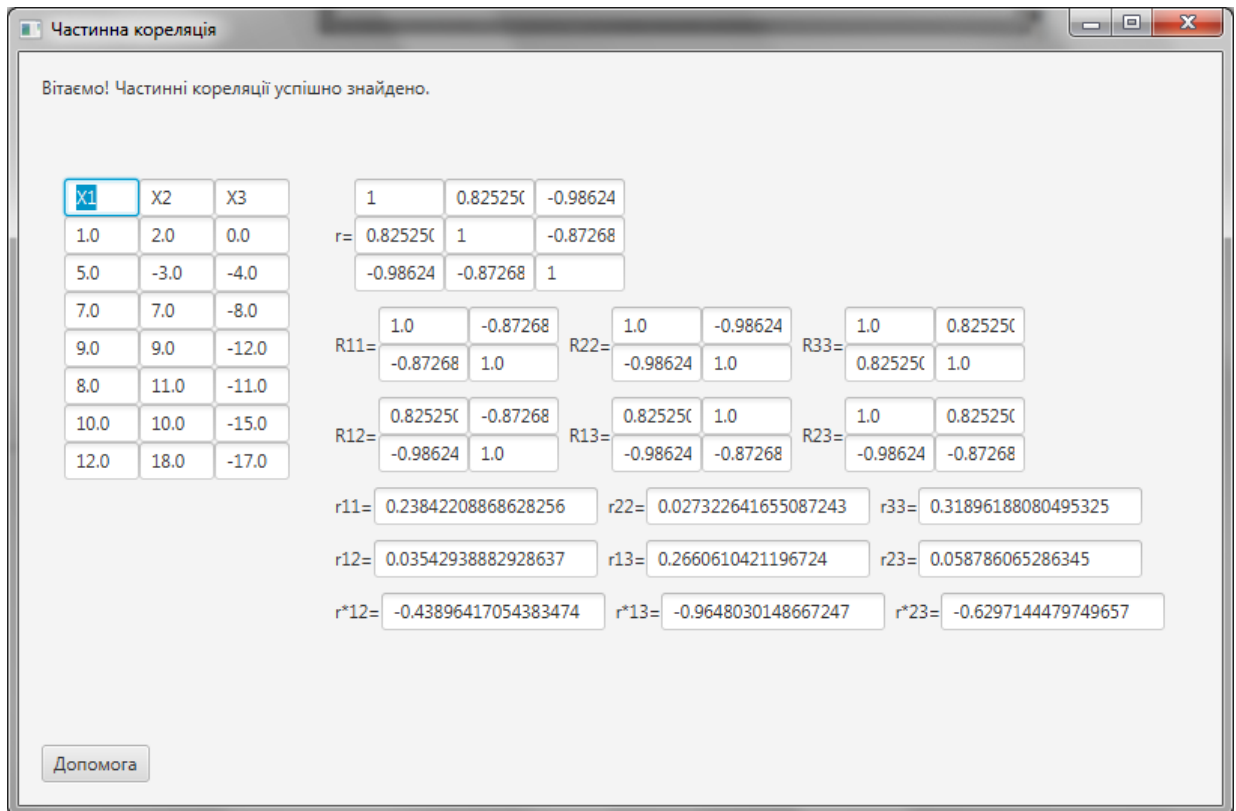


Рис. 4.26 Фінальне вікно після проходження тренажера

Отже, тестування тренажера пройшло успішно. Помилки і недоліків не виявлено.

ВИСНОВКИ

На кафедрі математичного моделювання та соціальної інформатики Полтавського університету економіки і торгівлі активно розробляються навчальні тренажери, які можуть використовуватися для дистанційного навчання та для інших форм навчання. Тренажери допомагають покращити практичну підготовку студентів з фахових предметів.

З дисципліни «Аналіз даних і прикладні пакети статистичної обробки» раніше розроблені навчальні тренажери з тем «Основні статистичні характеристики», «Кореляція якісних змінних», «Моделювання змінних», «Класифікація змінних», «Кореляція», «Парна лінійна регресія». Оскільки з теми «Частинна кореляція» тренажер ще не був розроблений, то тема бакалаврської роботи є актуальною.

Робота складається із вступу, чотирьох частин, висновків, списку літератури та додатку.

Головним результатом роботи є розробка тренажеру з теми «Частинна кореляція» навчальної дисципліни «Аналіз даних і прикладні пакети статистичної обробки».

Для розробки тренажеру використовувалася мова програмування Java. Для розробки графічного інтерфейсу користувача використана технологія JavaFX. Тренажер розроблений в інтегрованому середовищі розробки NetBeans.

Основні переваги створеного тренажеру:

- ✓ простота у використанні, зрозумілий інтерфейс;
- ✓ крос-платформність;
- ✓ можливість запуску тренажеру як із серверу, так і локально на комп'ютері користувача (при цьому доступ до мережі Internet не обов'язковий);
- ✓ незначне навантаження на сервер (у випадку запуску із серверу).

Результати роботи можуть використовуватися при вивченні студентами відповідної теми дисципліни «Аналіз даних і прикладні пакети статистичної обробки».

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Ємець О.О. Про розробку тренажерів для дистанційних курсів кафедрою ММСІ ПУЕТ / О.О. Ємець // Інформатика та системні науки (ІСН-2015): матеріали VI Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 19–21 берез. 2015 р.). – Полтава: ПУЕТ, 2015. – С. 152-161.
2. Ольховська О. В. Технології підтримки системи дистанційного навчання в Полтавському університеті економіки і торгівлі / О. В. Ольховська, Д. М. Ольховський // Інформатика та системні науки (ІСН-2016): матеріали VII Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 10–12 берез. 2016 р.). – Полтава: ПУЕТ, 2016. – С. 219-221.
3. Чілікіна Т. В. Огляд тренажерів з дисципліни "Математичний аналіз" на прикладі розробок студентів напряму "Інформатика" / Т. В. Чілікіна // Інформатика та системні науки (ІСН-2016): матеріали VII Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 10–12 берез. 2016 р.). – Полтава: ПУЕТ, 2016. – С. 329-330.
4. Черненко О. О. Методичні підходи щодо створення дистанційного курсу з дисципліни "Теорія програмування" / О. О. Черненко // Інформатика та системні науки (ІСН-2017): матеріали VIII Всеукраїнської науково-практичної конференції за міжнародною участю (м. Полтава, 16–18 березня 2017 р.) – Полтава: ПУЕТ, 2017. – С. 285-286.
5. Русін В. С. Програмна реалізація елементів тренажеру з теми "Аналіз алгоритму сортування вставками" дисципліни "Аналіз алгоритмів" / В. С. Русін, Ю. Ф. Олексійчук // Інформатика та системні науки (ІСН-2017): матеріали VIII Всеукраїнської науково-практичної конференції за міжнародною участю (м. Полтава, 16–18 березня 2017 р.) – Полтава: ПУЕТ, 2017. – С. 236-237.
6. Тюрин Ю.Н., Макаров А.А. Анализ данных на компьютере / По.ред. В.Э. Фигурнова. – М.: ИНФРА-М, Финансы и статистика, 1995. – 384 с.
7. Зюков М. Є. Аналіз даних: Навчально-методичний посібник. — Полтава: РВВ ПУСКУ, 2009. — 41 с.

8. Гайдышев И. Анализ и обработка данных: специальный справочник. — СПб: Питер, 2001. — 752 с.
9. Лялин В. С. Статистика: теория и практика в Excel. — М.: Финансы и статистика; ИНФРА-М, 2010. — 448 с.
10. Минько А.А. Статистический анализ в MS Excel. — М.: Издательский дом «Вильямс», 2004. — 448 с.
11. Берк К., Кэйри П. Анализ данных с помощью Microsoft Excel.: Пер. с англ. — М.: Издательский дом «Вильямс», 2005. — 560 с.
12. Эккель Б. Философия Java. Библиотека программиста. 4-е изд. / Б. Эккель — СПб: Питер, 2009. — 640 с.
13. Шилдт Г. Полный справочник по Java, 7-е издание./ Г. Шилдт — К.: Издательский дом "Вильямс", 2007. — 1034 с.
14. Машнин Т. С. JavaFX 2.0: разработка RIA-приложений / Т. С. Машнин — СПб.: БХВ-Петербург, 2012. — 320 с.
15. Монахов В. В. Язык программирования Java и среда NetBeans / В. В. Монахов — 3-е издание. — СПб.: «БХВ-Петербург», 2011. — 704 с.