

Харківський національний університет радіоелектроніки
Факультет Автоматики і комп'ютеризованих технологій

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки

Рівень вищої освіти перший (бакалаврський)

Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Автоматизація та комп'ютерно-інтегровані технології
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« 28 » квітня 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Кислому Богдану Івановичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення програмного забезпечення для визначення параметрів автоматизованих технологічних процесів для лабораторного практикуму студентів

затверджена наказом університету від 19 травня 2025 р. № 390Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 10 червня 2025 р.

3. Вихідні дані до роботи 3.1 Мова програмування – С#;

3.2 Фреймворк – Avalonia;

3.3 Тип інтерфейсу користувача – графічний;

3.4 Операційна система – Microsoft Windows 11 Pro;

3.5 Виведення значень обчислених коефіцієнтів;

3.6 Побудова графіків залежностей;

3.7 Виведення текстових результатів у вікні інтерфейсу після виконання відповідних розрахунків;

3.8 Виведення наборів параметрів через інтерфейс відповідно до обраної задачі.

4. Перелік питань, що потрібно опрацювати в роботі

4.1 Аналіз сучасних рішень для визначення параметрів автоматизованих технологічних процесів;

4.2 Аналіз потреб користувачів, для яких розробляється програмне забезпечення;

4.3 Збір інформації стосовно параметрів технологічних процесів;

4.4 Розроблення алгоритму програмного забезпечення;

4.5 Створення проєкту та репозиторію програмного забезпечення для обчислення параметрів автоматизованих технологічних процесів;

4.6 Тестування програмного забезпечення;

4.7 Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) _____
Слайди в форматі PowerPoint у кількості 10 слайдів з розширенням .pptx

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Аналіз технічного завдання	28.04.2025 – 30.04.2025	виконано
2	Опрацювання літератури за темою роботи	01.05.2025 – 03.05.2025	виконано
3	Аналіз сучасних рішень для визначення параметрів автоматизованих технологічних процесів	04.05.2025 – 12.05.2025	виконано
4	Розроблення програмного забезпечення	13.05.2025 – 27.05.2025	виконано
5	Оформлення пояснювальної записки	28.05.2025 – 01.06.2025	виконано
6	Подання роботи на перевірку Інтернет-сервісом StrikePlagiarism	02.06.2025 – 03.06.2025	виконано
7	Подання роботи на рецензію	04.06.2025 – 05.06.2025	виконано
8	Подання роботи на підпис зав.кафедри	06.06.2025 – 07.06.2025	виконано
9	Подання атестаційної роботи в ЕК	08.06.2025 – 09.06.2025	виконано

Дата видачі завдання 28 квітня 2025 р.

Здобувач _____

 (підпис)

Керівник роботи _____
 (підпис)

Богдан КИСЛИЙ
 (посада, власне ім'я, прізвище)
Ірина ЖАРІКОВА
 (посада, власне ім'я, прізвище)

Я, Кислий Богдан Іванович, як здобувач вищої освіти ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

01 червня 2025 р.



Богдан КИСЛИЙ

РЕФЕРАТ

Пояснювальна записка: 58 с., 5 табл., 14 рис., 5 дод., 23 джерел.

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ІНТЕРФЕЙС КОРИСТУВАЧА, ФРЕЙМВОРК AVALONIA, АВТОМАТИЗОВАНИЙ, ТЕХНОЛОГІЧНИЙ ПРОЦЕС, ЛАБОРАТОРНИЙ ПРАКТИКУМ.

Мета роботи – підвищення ефективності проведення лабораторного практикуму здобувачів освіти шляхом розроблення програмного забезпечення з відкритим вихідним кодом, що дозволяє виконувати обчислення параметрів автоматизованих технологічних процесів.

Об'єкт розробки – процес проведення інженерних обчислень, пов'язаних із визначенням характеристик автоматизованих технологічних систем у навчальному середовищі.

Предмет розробки – програмний засіб, що реалізує обчислювальні алгоритми для підтримки лабораторних занять з дисциплін, пов'язаних із автоматизацією та знаходженням параметрів автоматизованих технологічних процесів.

У роботі описано процес розробки програмного забезпечення для визначення параметрів автоматизованих технологічних процесів, які розраховуються студентами під час виконання лабораторних і практичних робіт. Проаналізовано перелік значень автоматизованих технологічних процесів, розрахунки яких повинні бути реалізовані в рамках застосунку, а також розроблено загальну структуру інтерфейсу користувача з урахуванням зручності та ефективності подання інформації.

Розроблено схему, алгоритм програмного забезпечення та інтерфейс, які дозволяють визначити параметри автоматизованих технологічних процесів та побудувати графіки залежностей між цими параметрами.

ABSTRACTS

The explanatory note contains 58 pages, 5 spreadsheets., 14 figures., 5 pp., 23 sources

SOFTWARE, PARAMETERS OF AUTOMATED TECHNOLOGICAL PROCESSES , C#, AVALONIA

Purpose of the work – to increase the efficiency of the students' laboratory workshop by developing open-source software that allows calculating the parameters of automated technological processes. Particular attention was paid to adapting the application to the specifics of the educational equipment used, as well as ensuring the required accuracy and reliability of the performed calculations.

Object of the development – the process of performing engineering calculations related to the determination of characteristics of automated technological systems in an educational environment.

Subject of the development – a software tool that implements computational algorithms to support laboratory classes in disciplines related to automation and parameter determination of automated technological processes.

This paper describes the process of developing software for determining parameters of automated technological processes studied by students during laboratory and practical assignments.

Typical problems in the field of automation were analyzed, a list of values of automated technological processes that should be calculated within the application was formed, and the general structure of the user interface was developed, taking into account convenience and efficiency in presenting information.

ЗМІСТ

Перелік скорочень	8
Вступ.....	9
1 Аналіз сучасних рішень для визначення ПАТП	11
1.1 Аналіз наукових публікацій	11
1.2 Патентний пошук	12
1.3 Актуальність роботи	15
1.4 Аналіз потреб користувачів, для яких розробляється ПЗ	16
1.5 Аналіз переваг і недоліків існуючих систем	17
2 Аналіз параметрів ТП і розрахункова частина	21
2.1 Аналіз загальних процесів роботи ПЗ.....	21
2.2 Збір інформації стосовно параметрів технологічних процесів	23
3 Реалізація програмного забезпечення	27
3.1 Аналіз мови програмування та фреймворку для розроблення ПЗ.....	27
3.2 Розроблення алгоритму ПЗ	28
3.3 Створення проекту та репозиторію ПЗ для обчислення ПАТП.....	31
3.4 Тестування ПЗ	36
3.4 Розроблення та інтеграція логотипу ПЗ в програмний код	47
3.5 Компіляція та публікація репозиторію на онлайн-платформу GitHub..	50
3.6 Охорона праці.....	51
Висновки	53
Перелік джерел посилання	55
Додаток А Апробація результатів роботи	59
Додаток Б Програмний код «MainWindow» для ПЗ.....	64
Додаток В Програмний код «CourseContent» для ПЗ.....	67
Додаток Г Налаштування C#-проєкту.....	96
Додаток Д Демонстраційний матеріал.....	97

ПЕРЕЛІК СКОРОЧЕНЬ

КІТАР – комп’ютерно-інтегровані технології, автоматизація та робототехніка;

КІТВЗА – комп’ютерно-інтегровані технології виробництва засобів автоматизації;

НСАВ – надійність системи автоматизованого виробництва;

ПАТП – параметри автоматизованих технологічних процесів для лабораторного практикуму студентів;

ПКЗЗ – програмно-керовані засоби зв’язку;

ПЗ – програмне забезпечення;

ПСА – проектування систем автоматизації;

САССУ – системний аналіз складних систем управління;

ТАУ – теорія автоматичного управління;

ТП – технологічний процес;

ХНУРЕ – Харківський національний університет радіоелектроніки.

ВСТУП

Під час виконання лабораторних і практичних робіт на дисциплінах спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології кафедри КІТАР, студенти все частіше й частіше стикаються з завданнями, в яких потрібно розрахувати різноманітні коефіцієнти, значення змінних, визначення параметрів автоматизованих технологічних процесів тощо. Більшість із них можна вирішити програмним шляхом. Виконаний алгоритм вирішуватиме потрібну задачу. Принцип алгоритму виконання програми незмінний:

- користувач вводить числа;
- користувач отримує результат, розрахований за допомогою підстановки введених чисел у формулу.

У кожній практичній та лабораторній роботах, де є розрахункова частина, завжди залишається можливість реалізовувати розв'язок інженерних виразів через програмний код (бо подекуди це може бути значно легше та швидше – задати алгоритм виконання, після чого вводити потрібні числа стільки разів, скільки це буде потрібно), замість того, щоб постійно переписувати одні й ті ж самі вирази з самого початку, для повторного розв'язання. Зокрема, це буде дуже ефективно у випадках, коли потрібно розрахувати одне й те ж саме значення декілька разів, не змінюючи при цьому формулу, але змінюючи введені значення.

Мета роботи – підвищення ефективності проведення лабораторного практикуму здобувачів освіти шляхом створення програмного забезпечення з відкритим вихідним кодом, яке дозволяє проводити обчислення параметрів автоматизованих технологічних процесів (ПАТП) у межах лабораторного практикуму студентів.

Об'єкт розробки – процес проведення інженерних обчислень, пов'язаних із визначенням характеристик автоматизованих технологічних систем у навчальному середовищі.

Предмет розробки – програмний засіб, що реалізує обчислювальні алгоритми для підтримки лабораторних занять з дисциплін, пов'язаних із автоматизацією та знаходженням параметрів автоматизованих технологічних процесів.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- розробити та організувати публічний репозиторій, у якому буде зберігатися програмне забезпечення (ПЗ), призначене для обчислення параметрів автоматизованих технологічних процесів у рамках лабораторних робіт здобувачів освіти;

- створити логотип і дизайн для програмного забезпечення;

- перевірити працездатність програмного забезпечення;

- викласти репозиторій на платформу GitHub [1] для можливості подальшого редагування похідного коду іншими користувачами та подальших оновлень ПЗ;

- оформити кваліфікаційну роботу згідно з ДСТУ 3008:2015 [2], а також з методичними вказівками з підготовки й оформлення кваліфікаційної роботи здобувачами першого (бакалаврського) рівня вищої освіти спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» [3], та посібником з підготовки кваліфікаційної роботи бакалавра для здобувачів вищої освіти [4].

Також матеріали досліджень щодо розроблюваного ПЗ були представлені на XXVIII Міжнародній науково-технологічній конференції «Технологія-2025» під час виступу з доповіддю та у вигляді наукової публікації у збірнику тез доповіді. Інформація щодо апробації результатів роботи представлена в Додатку А.

Проведені дослідження відповідають цілям сталого розвитку (ЦСР): ЦСР 4 та ЦСР 9.

1 АНАЛІЗ СУЧАСНИХ РІШЕНЬ ДЛЯ ВИЗНАЧЕННЯ ПАРАМЕТРІВ АВТОМАТИЗОВАНИХ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ

1.1 Аналіз наукових публікацій

У ході підготовки до виконання атестаційної роботи було проведено детальний аналіз наукових публікацій, технічної документації, а також прикладів схожих за принципом роботи програмних проєктів. Основна увага приділялась вивченню теоретичних і практичних напрацювань у галузі розроблення програмного забезпечення для інженерних розрахунків, яке застосовується в освітньому процесі, зокрема – у контексті проведення лабораторних практикумів та виконання навчальних завдань студентами спеціальності 151. Однією з ключових задач під час цього аналізу було виявлення сучасних підходів до автоматизації розрахункових процесів і пошук оптимальних рішень для реалізації подібного ПЗ.

На підставі огляду доступних публікацій було виявлено низку наукових статей, що акцентують увагу на актуальності впровадження цифрових засобів у навчальний процес. У таких джерелах підкреслюється важливість розвантаження студентів від рутинних і повторюваних обчислень, що не лише економить час, але й сприяє глибшому засвоєнню суті технологічних процесів і принципів функціонування автоматизованих систем. Також зазначається, що застосування програмного забезпечення дає можливість уникнути типових помилок, зумовлених людським фактором, особливо при багаторазовому виконанні ідентичних обчислень.

Особливий інтерес викликають публікації, присвячені використанню мов програмування C++, C# та Python у контексті створення навчальних обчислювальних інструментів. У цих роботах наголошується на доступності, кросплатформеності та високій продуктивності зазначених мов, що дозволяє створювати як консольні додатки, так і повноцінні графічні інтерфейси.

Зокрема, активно розглядаються бібліотеки для візуалізації графіків та побудови структурованих форм введення даних. Використання фреймворків цього типу дозволяє не лише реалізувати функціональну частину програмного продукту, а й забезпечити його зрозумілість для користувача – особливо важливо, коли мова йде про студентів без глибоких технічних знань у галузі програмування.

Окрім того, в окремих наукових роботах підкреслюється важливість відкритості програмного коду – як елементу академічної доброчесності та інструменту, що дозволяє іншим студентам або викладачам удосконалювати вже наявні рішення, адаптуючи їх під власні навчальні програми. Таким чином, відкрите ПЗ може стати універсальною платформою, яка постійно оновлюється відповідно до змін у методичних вказівках і програмі навчання.

Також було здійснено базовий патентний пошук із метою виявлення захищених авторським правом технічних рішень, які могли б перетинатися з ідеєю проєкту. Результати пошуку засвідчили, що хоча існують деякі загальні розробки для автоматизації обчислень у промисловості та освіті, прямої аналогії саме з програмним засобом, орієнтованим на лабораторний практикум у рамках певної спеціальності, виявлено не було. Це свідчить про доцільність реалізації власного унікального рішення, яке відповідає специфічним вимогам навчального процесу й може бути вдосконалене або адаптоване відповідно до майбутніх змін.

1.2 Патентний пошук

Патентний пошук проводився із залученням міжнародних баз даних (Google Patents, Espacenet, WIPO) та національних ресурсів. Приклади патентів зазначені в таблиці 1.1.

Таблиця 1.1 – Приклади патентів розробок

Номер патенту	Назва патенту	Опис патенту
CA2847769A1 [5]	Системи автоматизованого навчання з використанням декларативного підходу	Патент CA2847769A1 описує систему для вирішення задач у математиці, природничих науках та інженерії, яка використовує декларативний підхід до представлення рішень. Такий підхід дозволяє студентам краще розуміти процес розв'язання задач, що може бути корисним для автоматизації лабораторних робіт.
US8365138B2 [6]	Автоматизована система виробництва ПЗ	Патент US8365138B2 описує систему, яка автоматично генерує повноцінне ПЗ на основі формальних специфікацій. Така система може бути адаптована для створення навчальних програм, що автоматизують розрахунки в лабораторних роботах.
US20040133532A1 [7]	Системи комп'ютерного навчання	Патент US20040133532A1 описує системи комп'ютерного навчання, які надають студентам педагогічно відповідну взаємодію, включаючи визначення інформації про учня та створення відповідних шаблонів взаємодії. Такий підхід може бути використаний для адаптації навчального програмного забезпечення до потреб студентів.

У процесі підготовки атестаційної роботи було здійснено патентний пошук з метою виявлення наявних технічних рішень, пов'язаних із темою розроблення ПЗ для визначення ПАТП у контексті освітнього застосування.

Основна увага приділялася розробкам, які охоплюють створення інженерних калькуляторів, автоматизованих програмно-методичних комплексів, що використовуються для навчання студентів технічних спеціальностей, а також систем підтримки прийняття рішень, призначених для використання у виробничих або лабораторних умовах.

У ході аналізу патентних баз було виявлено значну кількість рішень прикладного характеру, які, як правило, мають вузьку спеціалізацію. Наприклад, деякі з них спрямовані на розрахунок теплових режимів окремого обладнання, розрахунок параметрів пневматичних або гідравлічних систем, визначення навантажень у конструкціях тощо. Водночас ці рішення розроблялися переважно під конкретні виробничі задачі або під специфічні умови експлуатації, тому їх застосування у контексті освітніх лабораторних практикумів є або неможливим, або малоефективним без суттєвої адаптації.

Слід також зазначити, що у виявлених патентах практично відсутні рішення, спеціально орієнтовані на підтримку методичних вказівок до лабораторних і практичних робіт із технологічних дисциплін. У більшості випадків не розглядається можливість інтеграції подібного ПЗ у навчальні плани або адаптації під різні формати освітніх програм, що суттєво обмежує їх потенціал для використання у вищій технічній освіті. Це підтверджує актуальність і доцільність розроблення нового програмного продукту, адаптованого саме для потреб освітнього процесу.

Окрему увагу під час пошуку було приділено дотриманню вимог інтелектуальної власності. Розроблюване програмне забезпечення створюється з нуля, із застосуванням виключно авторського коду, без використання сторонніх бібліотек, ресурсів, логотипів чи зображень, що можуть бути захищені авторським правом. Такий підхід забезпечує юридичну прозорість продукту та дозволяє безперешкодно використовувати його в освітньому середовищі, зокрема в умовах відкритого доступу до вихідного коду.

Таким чином, ідея атестаційної роботи, яка полягає у створенні ПЗ для автоматизованого визначення параметрів технологічних процесів у рамках лабораторного практикуму студентів, не має прямих аналогів серед запатентованих розробок, що підкреслює її новизну. Відкритий характер розробки, передбачений проектом, дозволяє у майбутньому доповнювати функціонал, адаптувати програму до інших дисциплін або специфіки технічного обладнання, що використовується у закладі освіти. Це робить програмний засіб потенційно корисним не лише в межах окремого навчального курсу, а й як частину ширшої цифрової освітньої екосистеми.

1.3 Актуальність роботи

Сучасна освітня практика у сфері автоматизації технологічних процесів (ТП) та програмування передбачає активне використання інженерних розрахунків під час виконання лабораторних і практичних робіт. Однак, у більшості випадків, ці розрахунки виконуються вручну або з використанням універсальних калькуляторів, що не враховують специфіку методичних вказівок і не дозволяють легко адаптуватися до змін навчального матеріалу.

У той же час, студенти протягом навчання набувають навичок програмування, що відкриває можливість створення власних програмних засобів для спрощення розрахункової частини робіт. Розробка спеціалізованого ПЗ, яке автоматизує рутинні розрахунки згідно з методиками навчальних дисциплін, дозволяє не лише підвищити ефективність виконання завдань на лабораторні роботи, а й краще принцип розв'язання інженерних задач.

Особливої актуальності набуває створення універсального та відкритого програмного рішення, адаптованого до конкретної спеціальності. Таке ПЗ дає змогу студентам вводити необхідні параметри, отримувати результати в зручній формі, а також легко редагувати або розширювати програму відповідно до оновлень методичних вказівок. Відкритий похідний код,

розміщений у загальнодоступному репозиторії, забезпечує гнучкість і можливість колективної розробки.

Таким чином, створення ПЗ для визначення ПАТП не лише відповідає сучасним вимогам до освітнього процесу, а й сприяє підвищенню якості інженерної підготовки студентів.

1.4 Аналіз потреб користувачів, для яких розробляється програмний засіб

Програмне забезпечення розробляється насамперед для студентів технічних спеціальностей, які під час навчання виконують лабораторні та практичні роботи з різних дисциплін, пов'язаних із автоматизацією, програмуванням та іншими подібними напрямками. Дуже часто у таких роботах виникає потреба у виконанні великої кількості однотипних розрахунків. При цьому формули можуть бути складними, містити багато параметрів і потребувати багаторазового ручного введення одних і тих самих даних. Через це легко припуститися помилки, особливо коли потрібно кілька разів перераховувати результати з різними вихідними значеннями.

У великій кількості випадків, доводиться вручну вписувати одні й ті самі вирази або використовувати сторонні інструменти, які не враховують специфіку методичних вказівок конкретної дисципліни.

По цим причинам, студентам потрібен зручний інструмент, який дозволить легко й швидко підставляти потрібні значення в формули, отримувати результати без повторень, і при цьому – мати можливість за потреби гнучко змінювати саму структуру обчислень. Ще одна перевага даного програмного засобу – це зрозумілий інтерфейс, у якому можна буде кілька разів вводити різні дані для одного й того ж розрахунку, не переписуючи все спочатку.

Ця програма може стати корисною не тільки для студентів, а й для викладачів. Вони зможуть використовувати її для перевірки типових завдань, створення прикладів, а також для демонстрації, як працює той чи інший алгоритм прямо під час занять.

Загалом, потреба в такому програмному засобі виникає з практики самого навчального процесу. Ідея полягає в тому, щоби максимально полегшити рутинну частину роботи, щоби залишити більше часу на розуміння самої суті розрахунків, а не витратити години на ручні підстановки та перевірки.

1.5 Аналіз переваг і недоліків існуючих систем

Було проведено аналіз потреб і вимог користувачів, для яких розробляється ПЗ, а також проаналізовано переваги та недоліки популярних процесів і систем для визначення параметрів. Результати аналізу записані в таблицю 1.2.

Таблиця 1.2 – Переваги та недоліки популярних процесів і систем для визначення параметрів

Процес / система	Переваги	Недоліки
Калькулятори	1. Зручність 2. Відносна гнучкість у використанні 3. Стабільність	1. Для повторних розрахунків треба повторного переписувати одні і ті самі формули 2. Кожна з наступних формул записується власноруч

Продовження таблиці 1.2

Процес / система	Переваги	Недоліки
Власні програмні засоби	<ol style="list-style-type: none"> 1. Максимально гнучкий функціонал, оскільки його обсяги та стабільність ПЗ залежать від студента 2. Не потребують постійного переписування подібних формул при повторенні розрахунків 	<ol style="list-style-type: none"> 1. Початково, їх не існує. Потребують створення студентом 2. Є ймовірність допустити похибку в створенні формули, що залежить від досвіду та уважності студента
Письмові розрахунки	<ol style="list-style-type: none"> 1. Не потребують використання комп'ютерів, телефонів і калькуляторів 2. Припустимі для використання на іспитах 	<ol style="list-style-type: none"> 1. Після кожних змін, потрібно переписувати формулу власноруч 2. Дуже незручні та повільні 3. Велика ймовірність помилки через людський фактор

У процесі аналізу існуючих рішень для визначення ПАТІ було розглянуто три основні підходи: використання калькуляторів, власних програмних засобів і письмових розрахунків. Кожен із них має свої переваги та недоліки, що безпосередньо впливають на зручність, точність та ефективність виконання обчислень у навчальному процесі.

Калькулятори залишаються одним із найбільш популярних і зручних інструментів завдяки своїй стабільності та відносній гнучкості у використанні. Вони дозволяють швидко вводити числові значення й отримувати результат, не потребуючи складних налаштувань або додаткового програмування. Однак при повторенні розрахунків виникає необхідність кожного разу самостійно переписувати формули вручну, що не лише знижує ефективність, а й підвищує ризик помилок.

Одним із найкращих прикладів сучасних калькуляторів є модульний калькулятор Photomath. Зображення з його використанням наведено на рисунку 1.1.



Рисунок 1.1 – Використання програми «Photomath» [8]

Студенти також можуть розробляти власні програмні проекти, які можуть забезпечити максимальну гнучкість і адаптованість до конкретних потреб. Вони дозволяють зберігати структуру формул, автоматизувати процес введення змінних і отримання результатів, а також уникати зайвого переписування одних і тих самих виразів. Такий підхід значно заощадить час і знизить імовірність помилок під час повторних обчислень. Проте основним недоліком є те, що такі інструменти потребують створення студентом, що вимагає певних навичок програмування та затрат часу. Окрім того, на етапі формування функціоналу існує ризик допущення додаткових помилок, які залежать від уважності та компетентності розробника.

Письмові розрахунки є найбільш традиційним підходом і не потребують жодних технічних засобів. Їх можна використовувати навіть під час очних іспитів, де заборонено застосовувати електронні пристрої. Водночас цей метод вважається найбільш незручним і трудомістким. Кожна зміна вхідних параметрів змушує переписувати всю формулу з нуля, що забирає багато часу

та енергії. Також суттєвим недоліком є висока ймовірність помилок через людський фактор, особливо в умовах обмеженого часу та/або втоми.

Загалом, аналіз показує, що найперспективнішим напрямом є створення власного програмного забезпечення, яке дозволяє об'єднати зручність автоматизації з гнучкістю налаштування під специфічні навчальні потреби. Попри те, що його впровадження потребує початкових зусиль, такі рішення у довгостроковій перспективі є найбільш ефективними та корисними для студентів.

2 АНАЛІЗ ПАРАМЕТРІВ ТП І РОЗРАХУНКОВА ЧАСТИНА

2.1 Аналіз загальних процесів роботи ПЗ

На основі проаналізованих процесів і систем для визначення параметрів, було побудовано діаграму IDEF0, яка дозволяє представити основні процеси обробки даних, їхню взаємодію та зв'язки із зовнішніми джерелами. Результат показаний на рисунку 2.1.

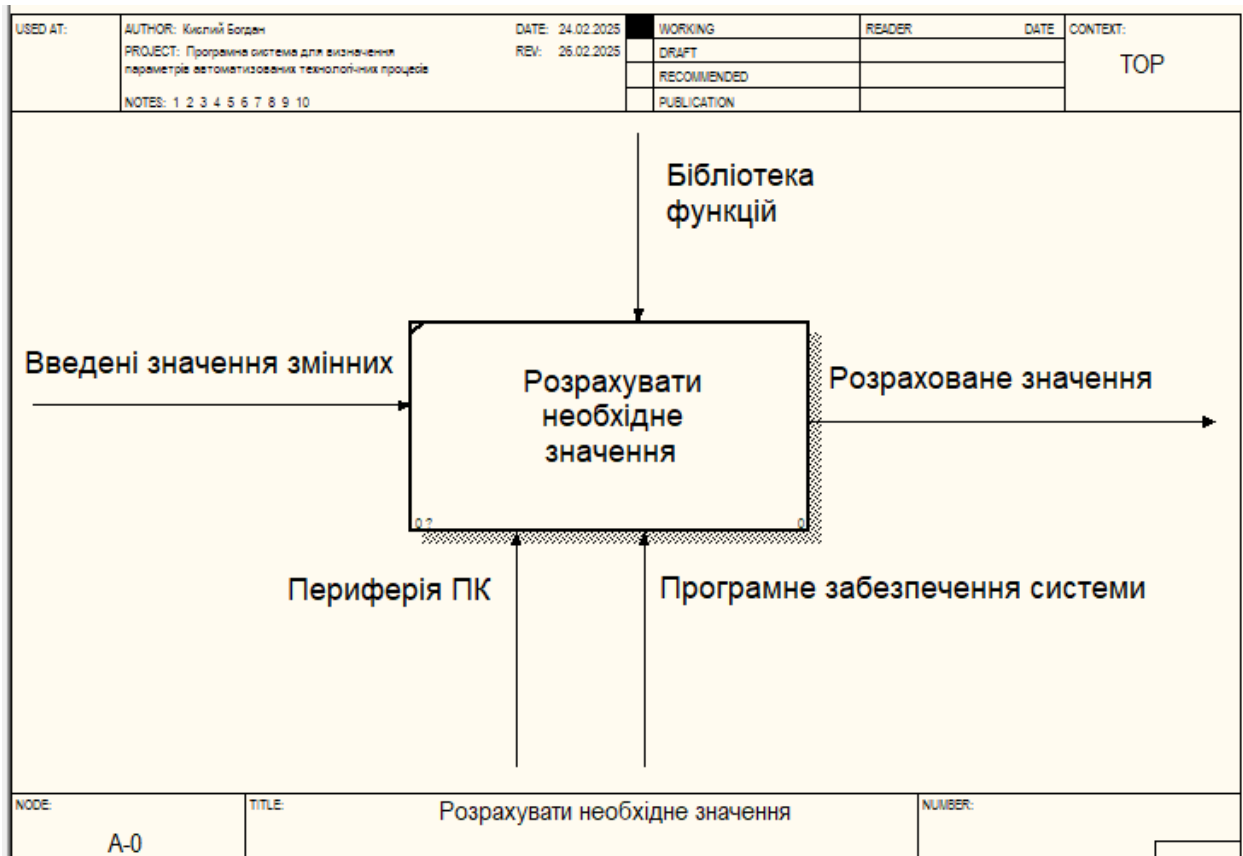


Рисунок 2.1 – Функціональна діаграма ПЗ

На першому рівні функціональної діаграми розглянуто загальний процес «Розрахувати необхідне значення». Вхідними даними для нього є змінні, введені користувачем, а виходом – розраховане значення відповідно до заданої формули. Серед керуючих впливів виділено бібліотеку функцій та програмне

забезпечення системи, а безпосереднє виконання відбувається з використанням апаратних засобів персонального комп'ютера.

Для опису логіки роботи програмного забезпечення та подальшого планування його реалізації, була проведена декомпозиція діаграми рівня А-0. Головну функцію системи можна умовно поділити на п'ять, пов'язаних із вибором формули, введенням значень змінних і використанням їх у обраній формулі, а також із розв'язанням задачі та отриманням значення (рисунок 2.2).

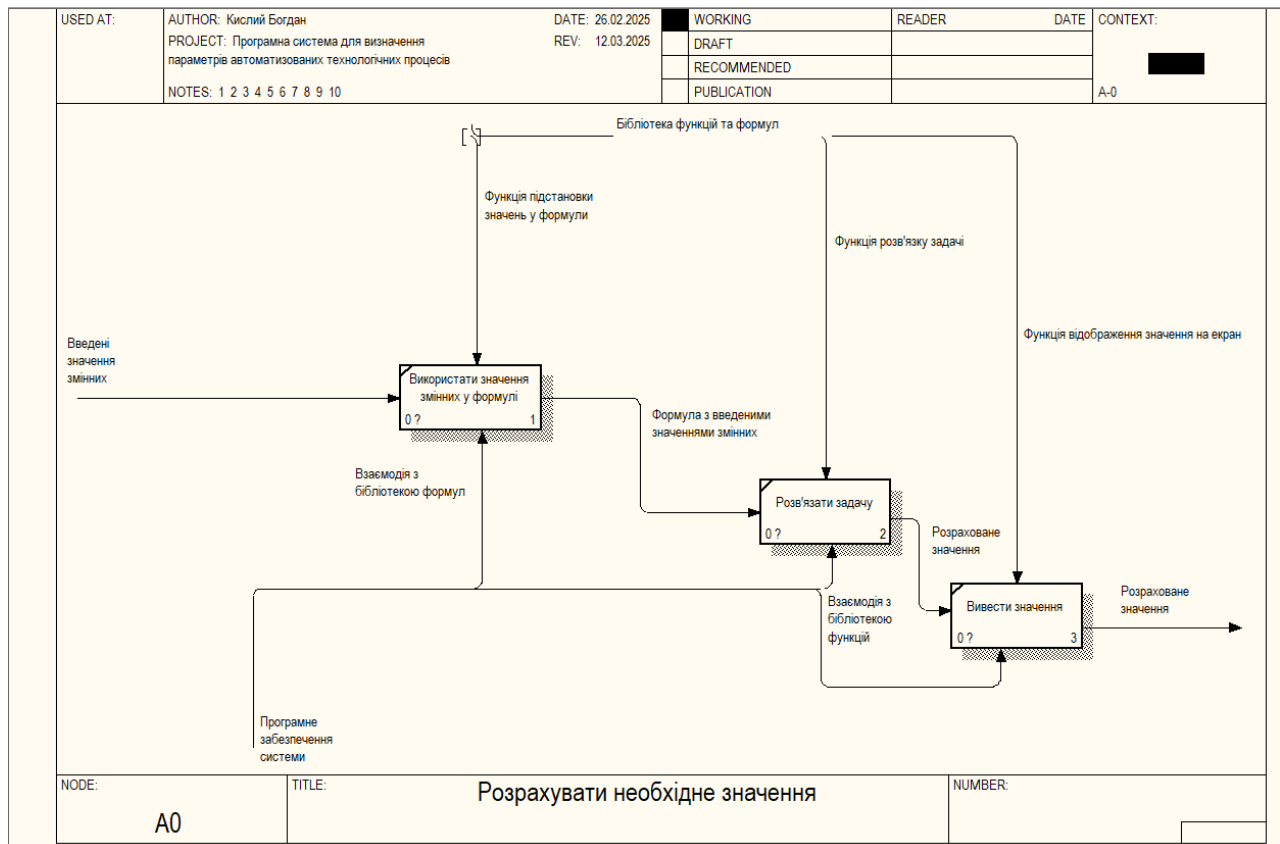


Рисунок 2.2 – Декомпозиція рівня А-0 функціональної діаграми ПЗ

На наступному рівні деталізації (А0) представлено декомпозицію цього загального процесу на окремі функціональні блоки. Уся логіка реалізована таким чином, щоб зменшити кількість ручної роботи для користувача, зменшити ймовірність помилок та надати швидкий і точний результат.

Для кращого розуміння логіки функціонування було також розроблено дерево вузлів, яке відображає ієрархію завдань, що виконуються під час роботи програми.

Основний вузол – «Розрахувати необхідне значення» – має кілька дочірніх елементів, які відповідають ключовим етапам, зазначеним на рисунку 2.3.



Рисунок 2.3 – Дерево вузлів функціональної моделі програмної системи

2.2 Збір інформації стосовно параметрів технологічних процесів

Перед початком розроблення ПЗ, варто визначити, які саме ПАТП воно зможе розрахувати. Скористаємося методичними вказівками з таких дисциплін, як «Фізика» [9], «Системний аналіз складних систем управління» [10], «Теорія автоматичного управління» [11], «Комп'ютерно-інтегровані технології виробництва засобів автоматизації» [12] та «Проектування систем автоматизації» [13].

Усі параметри з загальною інформацією записані у вигляді таблиці 2.1.

Таблиця 2.1 – Параметри для знаходження

Параметр	Дисципліна	Курс	Формули
Густина полого циліндра	Фізика	1	$\rho = \frac{4m}{\pi h(D^2 - d^2)}$
Густина суцільного циліндра	Фізика	1	$\rho = \frac{4m}{\pi h d^2}$
Густина пластинки	Фізика	1	$\rho = \frac{m}{abc}$
Густина кулі	Фізика	1	$\rho = \frac{6m}{\pi D^3}$
Модель максимальної узагальненої корисності	САССУ	2	$P(x) = \max_{x \in X} \sum_{i=1}^n a_i p_i(x)$
Критерій максимального математичного сподівання	САССУ	2	$E(x^*) = \max_{x \in X} M(x)$
Критерій мінімальної дисперсії	САССУ	2	$E(x^*) = \min_{x \in X} D(x)$
Критерій «очікуване значення – дисперсія»	САССУ	2	$E(x^*) = \max_{x \in X} (M(x) - K * D(x))$
Асимптотична частотна характеристика	ТАУ	3	$W(s) = \frac{k(T_1 s + 1)(T_2^2 s^2 + T_2 s + 1)}{s^v (T_3 s + 1)(T_4^2 s^2 + T_4 s + 1)}$
Кореневий критерій	ТАУ	3	$as^2 + bs + c \Rightarrow \begin{cases} s_1 \\ s_2 \end{cases}$
Критерій Лапласа	ТАУ	3	$as^2 + bs + c \Rightarrow \begin{vmatrix} a & c \\ 0 & b \end{vmatrix}$
Коефіцієнт серійності	КІТВЗА	3	$K_c = \frac{K_o}{n_p}$

Продовження таблиці 2.1

Параметр	Дисципліна	Курс	Формули
Кількість робочих місць, необхідних для виконання процесу виготовлення виробу	КІТВЗА	3	$n_p = \frac{N \sum_{i=1}^{K_o} T_{um.i}}{60 \cdot K \cdot \Phi_\partial}$
Комплексний показник технологічності	КІТВЗА	3	$K = \frac{\sum_{i=1}^7 K_i \phi_i}{\sum_{i=1}^7 \phi_i}$
Надійність системи автоматизованого виробництва (НСАВ) за методом Шумана та Лямбда	ПСА	4	$\varepsilon_c(\tau_A) = \frac{x_1 + x_2 + x_3}{I},$ $\varepsilon_c(\tau_B) = \frac{x_4 + x_5 + \dots + x_9 + x_{10}}{I},$ $\lambda_A = \frac{N_A}{\tau_A}, \lambda_B = \frac{N_B}{\tau_B},$ $C = \frac{\lambda_A}{\frac{N_0}{I} - \varepsilon_c(\tau_A)},$ $R(t, \tau) = e^{-C[\frac{N_0}{I} - \varepsilon_c(x)]t}$
НСАВ за методом Мілса	ПСА	4	$N_n = \frac{W_n * S_n}{V_n},$ $r_n = r_{n-1} - S_{n-1},$ $W_n = W_{n-1} - V_{n-1},$ $C = \frac{W}{W + r + 1}$ <p>де n – номер прогону</p>

Продовження таблиці 2.1

Параметр	Дисципліна	Курс	Формули
НСАВ за моделлю Джелінскі- Моранді	ПСА	4	$(k - 1) \times \frac{\sum_{i=1}^{k-1} t_i}{\sum_{i=1}^{k-1} \frac{1}{N - i + 1}},$ $\sum_{i=1}^{k-1} (N - i + 1)t_i,$ $C_D = \frac{\sum_{i=1}^{k-1} \frac{1}{N - i + 1}}{\sum_{i=1}^{k-1} t_i},$ $\lambda_i = C_D(N - (i - 1)) \Rightarrow t = \frac{1}{\lambda_i},$
НСАВ за моделлю Нельсона	ПСА	4	$R = 1 - \sum_{i=1}^N P_i \times y_1$

У ПЗ будуть передбачені функції та методи для розрахунків представлених параметрів автоматизованих технологічних процесів, а також побудова графіків залежностей.

3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Аналіз мови програмування та фреймворку для розроблення програмного забезпечення

Мовою програмування для розроблення ПЗ для розрахунку ПАТП була обрана мова C#, оскільки існує щонайменше 2 відомих та відносно зручних фреймворки, які дозволяють створити не тільки ПЗ, але і UI для цього ПЗ.

Найкращі варіанти – це «Avalonia» [14] та «Windows Forms» [15]. З метою обрати найкращий для розроблюваної програми варіант, було проведено порівняння цих двох фреймворків, та занесене в таблицю 3.1.

Таблиця 3.1 – Порівняння фреймворків «Avalonia» та «Windows Forms»

Фреймворк	Переваги	Недоліки
Avalonia	<ol style="list-style-type: none"> 1. Кросплатформенність (Windows, Linux, macOS) 2. Сучасний та адаптивний інтерфейс 3. Повна підтримка MVVM 4. Висока гнучкість та кастомізація 5. Активна підтримка та розвиток 	<ol style="list-style-type: none"> 1. Відсутність drag-and-drop редактора 2. Вищий поріг входу 3. Менше туторіалів 4. Необхідність ручного налаштування кросплатформеності
Windows Forms	<ol style="list-style-type: none"> 1. Простота для початківців 2. Швидке розроблення через drag-and-drop 3. Готове для Windows 4. Велика база прикладів та рішень 	<ol style="list-style-type: none"> 1. Застарілий вигляд 2. Працює лише на Windows 3. Обмежені можливості стилізації 4. Слабка підтримка MVVM без додаткового коду

У результаті порівняння двох популярних технологій розробки інтерфейсу користувача – Windows Forms та Avalonia – було виявлено, що хоча Windows Forms є простішим у реалізації та добре підходить для створення базових програм під Windows, його функціональні можливості та візуальна складова обмежені, а сама технологія поступово втрачає актуальність. Натомість Avalonia, хоч і потребує трохи більше зусиль для освоєння, пропонує сучасний адаптивний інтерфейс, повну підтримку шаблону MVVM та головне – можливість створення кросплатформеного програмного забезпечення, що є важливою перевагою у сучасних умовах.

Враховуючи ці фактори, для реалізації програмного забезпечення в межах атестаційної роботи було обрано Avalonia як більш перспективну та гнучку платформу, яка дозволяє створити зручний, масштабований і придатний для подальшого розвитку продукт.

3.2 Розроблення алгоритму ПЗ

У програмному забезпеченні алгоритми відіграють ключову роль, оскільки вони визначають, яким чином програма реагує на введення даних, приймає рішення, обчислює результати та виводить їх користувачеві. Алгоритм реалізується у вигляді реакції на вибір курсу та завдання, відкриття відповідного вікна, введення коефіцієнтів, запуску розрахунку та виводу результату. Логіка роботи оформлена у вигляді структурованого коду, який виконує обчислення на основі даних користувача.

Алгоритм роботи програми включає наступні кроки:

- користувач запускає головне вікно програми;
- користувач обирає курс;
- програма відображає список доступних завдань для цього курсу;
- користувач натискає на назву завдання;
- відкривається нове вікно для введення необхідних коефіцієнтів;

- користувач вводить значення у відповідні поля;
- користувач натискає кнопку «Розрахувати!»;
- програма обробляє введені дані, викликаючи відповідну функцію розрахунку;
- результат обчислення виводиться на екран та автоматично зберігається в буфері обміну для подальшої обробки.

На рисунку 3.1 представлена діаграма виду «ACTIVITY DIAGRAM» алгоритму роботи програмного забезпечення. Вона демонструє основні етапи функціонування системи – від початкової взаємодії з користувачем до виконання обчислень і виведення результату.

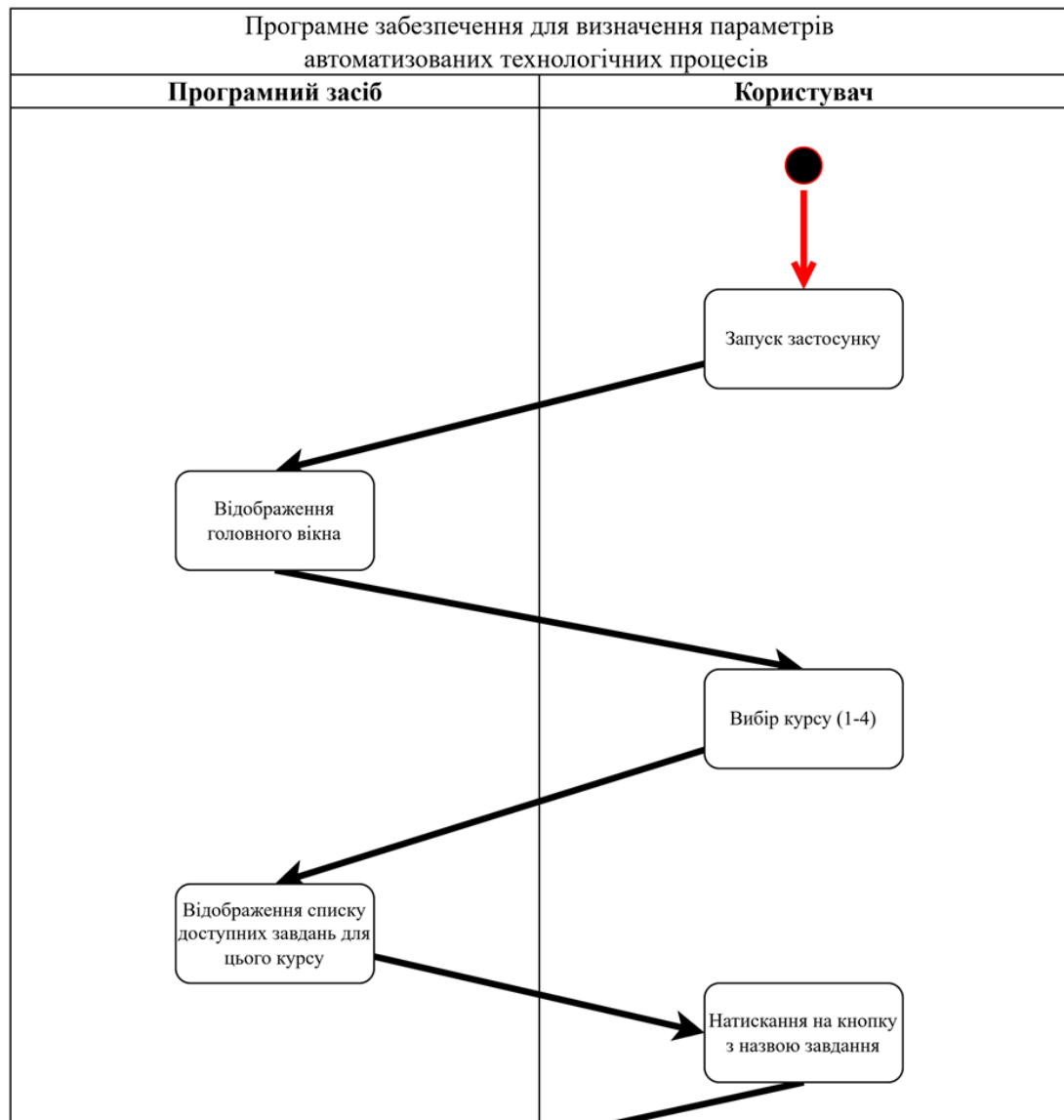


Рисунок 3.1 – Алгоритм роботи ПЗ

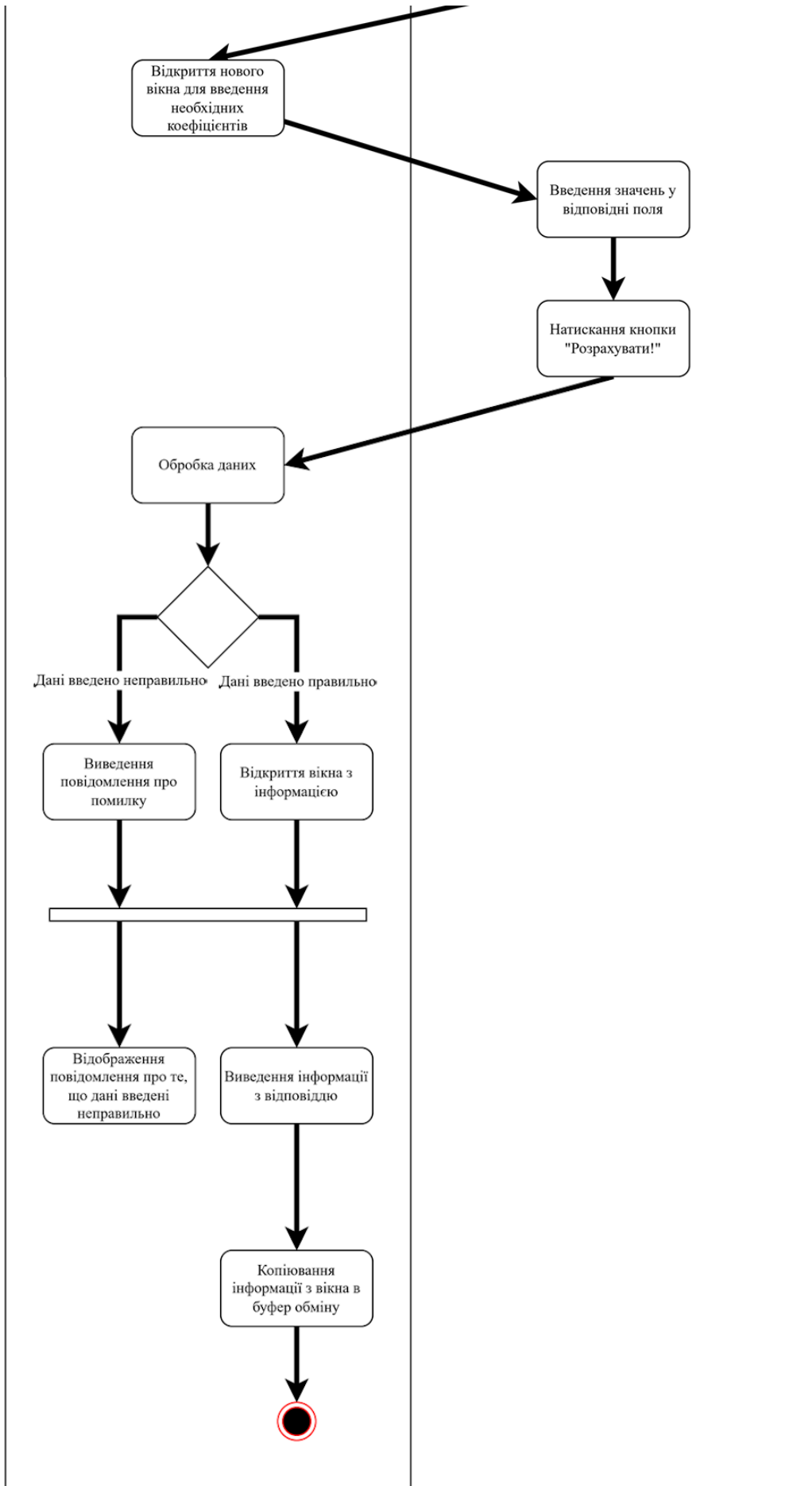


Рисунок 3.1, аркуш 2

Вся логіка обробки даних, перевірки введених значень і реалізації математичних формул реалізована за допомогою структурованого коду, написаного мовою C# у рамках архітектури Avalonia UI.

3.3 Створення проекту та репозиторію ПЗ для обчислення ПАТП

Для створення програми використаємо IDE під назвою Microsoft Visual Studio 2022 Community та фреймворк «Avalonia». Відкривмо Visual Studio та створимо проєкт (рисунок 3.2). Як шаблон обираємо «Avalonia .NET MVVM App (AvaloniaUI)».

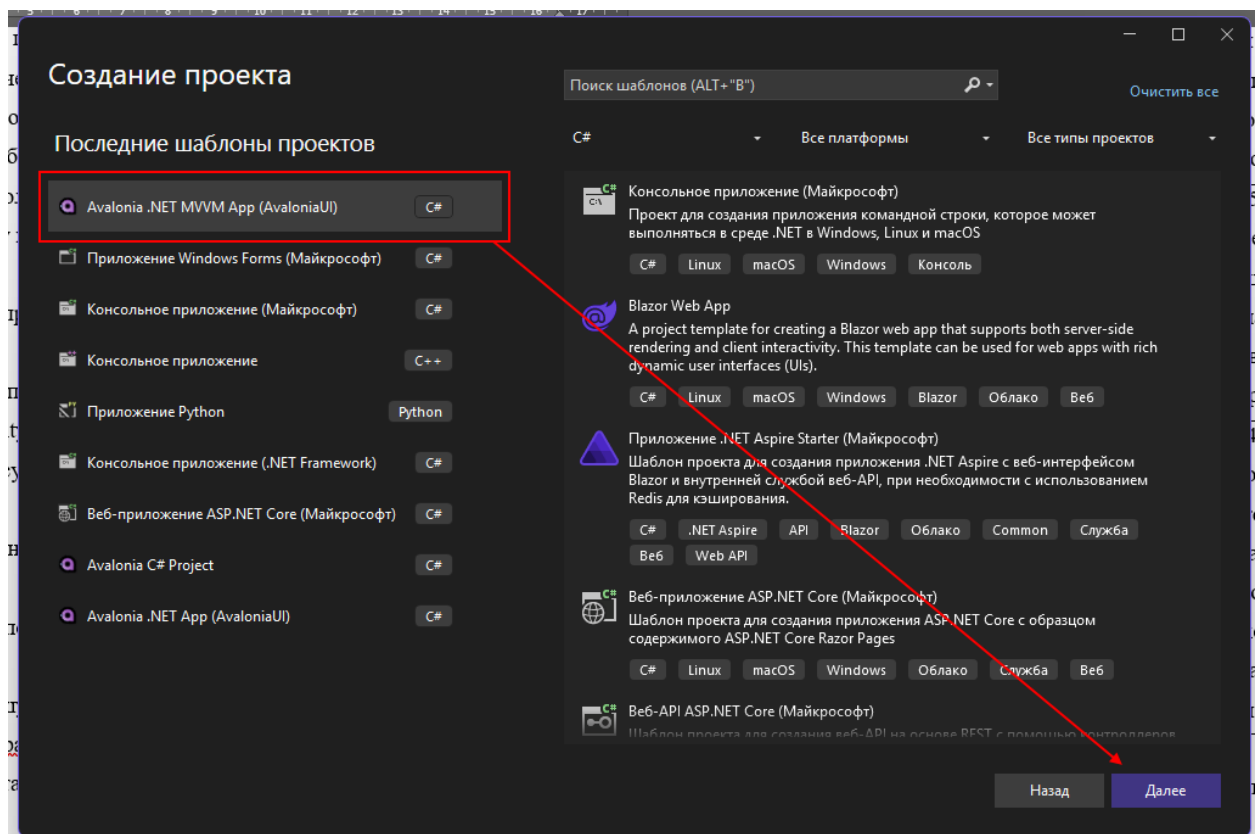


Рисунок 3.2 – Створення проєкту в Visual Studio

Буде автоматично створений файл `MainWindow.axaml`. У нього запишемо наступний код (повний код буде доступний в додатку Б):

```
<Border Grid.Column="0" Background="#2E3A59" Padding="10">
```

```

<StackPanel Spacing="10">
    <TextBlock Text="Оберіть курс:" FontWeight="Bold"
Foreground="White"/>
    <Button Content="1 курс" Command="{Binding
SelectCourseCommand}" CommandParameter="1"/>
    <Button Content="2 курс" Command="{Binding
SelectCourseCommand}" CommandParameter="2"/>
    <Button Content="3 курс" Command="{Binding
SelectCourseCommand}" CommandParameter="3"/>
    <Button Content="4 курс" Command="{Binding
SelectCourseCommand}" CommandParameter="4"/>

```

Розробимо ViewModel для цього файлу, та приєднаємо до нього новоутворений клас «CourseContentView». Весь код запишемо в файл MainWindowViewModel.cs:

```

public class MainWindowViewModel : ViewModelBase
{
    private object? _selectedCourseContent;
    public object? SelectedCourseContent
    {
        get => _selectedCourseContent;
        set => this.RaiseAndSetIfChanged(ref _selectedCourseContent,
value);
    }
    private void OnCourseSelected(string course)
    {
        SelectedCourseContent = new CourseContentView { DataContext =
new CourseContentViewModel(course) };
    }
}

```

Після цього розробимо область головного вікна для відображення контенту після вибору курсу, та винесемо всю верстку в файл

CourseContentView.axaml. У нього запишемо наступний код (повний код буде доступний в додатку В):

```
<StackPanel Margin="20" Spacing="10">
    <!--<views:CourseContentView />-->
    <TextBlock Text="{Binding CourseName}" FontSize="20"
FontWeight="Bold"/>
    <ItemsControl ItemsSource="{Binding Buttons}">
        <ItemsControl.ItemTemplate>
            <DataTemplate>
                <Button Content="{Binding}" Margin="0,5,0,0"
Click="Button_Click"/>
            </DataTemplate>
        </ItemsControl.ItemTemplate>
    </ItemsControl>
</StackPanel>
```

Також запишемо всю необхідну інформацію в автоматично створений файл CourseContentView.axaml.cs:

```
void Main_HollowCylinder(double m, double h, double D, double d)
[...]
void Main_SolidCylinder(double m, double h, double d) [...]
void Main_BallDensity(double m, double D) [...]
```

У цьому фрагменті знаходяться методи для знаходження густини полого та суцільного циліндрів, а також густини пластинки та густини кулі. Це параметри, які потрібно знайти в лабораторних роботах з фізики на першому курсі. У наступному фрагменті продемонстровані методи для знаходження ПАТТ з дисциплін «САСС», «ТАУ» та «КІТВЗА»:

```
void Main_CriterionOfMaximumMathematicalExpectation(double
M_x1, double M_x2, double M_x3, double M_x4) [...]
void Main_CriterionOfMinimumDispersion(double D_x1, double D_x2,
double D_x3, double D_x4) [...]
```

```
void Main_ExpectationDispersion(double K, double M_x1, double
M_x2, double M_x3, double M_x4, double D_x1, double D_x2, double D_x3,
double D_x4) [...]
```

```
void Main_AsymptoticFrequencyResponse(double k, double T1,
double T2, double T3, double T4, double s, double v) [...]
```

```
void Main_RootAndLaplasCriterion(double a, double b, double c,
double s) [...]
```

```
void Main_SerializationRate(double K_O, double n_p) [...]
```

```
void Main_ComplexManufacturabilityIndicator(double K_i, double F_i)
[...]
```

У наступному фрагменті продемонстровані методи для знаходження ПАТП з дисципліни «ПСА»:

```
void Main_NSAV1(double x1, double x2, double x3, double x4, double
x5, double x6, double x7, double x8, double x9, double x10, double N_A, double
T_A, double N_B, double T_B, double O, double E_C, double I, double t, double x)
[...]
```

```
void Main_NSAV2(double W_1, double S_1, double V_1, double W_2,
double S_2, double V_2, double W_3, double S_3, double V_3, double W_4, double
S_4, double V_4, double W_5, double S_5, double V_5, double W_6, double S_6,
double V_6) [...]
```

```
void Main_NSAV3(double k, double N, double t1, double t2, double t3,
double t4, double i1, double i2, double i3, double i4, double i) [...]
```

```
void Main_NSAV4(double p_1, double p_2, ... , double p_29, double
p_30, double y_1, double y_2, ... , double y_29, double y_30) [...]
```

Також додаємо нові поля меню для відображення – для цього створимо ще один ViewModel у новому С#-файлі під назвою CourseContentViewModel.cs. Також створимо в ньому нову функцію автоматизованого відображення нових параметрів для розрахунків:

```
switch (course)
```

```

{
    case "1":
Buttons.Add("Густина полого циліндра");
Buttons.Add("Густина суцільного циліндра");
Buttons.Add("Густина пластинки");
Buttons.Add("Густина кулі");
break;
    case "2":
Buttons.Add("Модель максимальної узагальненої корисності");
Buttons.Add("Критерій максимального математичного сподівання");
Buttons.Add("Критерій мінімальної дисперсії");
Buttons.Add("Критерій «очікуване значення – дисперсія»");
break;
    case "3":
Buttons.Add("Асимптотична частотна характеристика");
Buttons.Add("Кореневий критерій та критерій Лапласа");
Buttons.Add("Коефіцієнт серійності");
Buttons.Add("Кількість робочих місць, необхідних для виконання\процесу
виготовлення виробу");
Buttons.Add("Комплексний показник технологічності");
break;
    case "4":
Buttons.Add("НСАВ за методом Шумана та Лямбда");
Buttons.Add("НСАВ за методом Мілса");
Buttons.Add("НСАВ за моделлю Джелінські-Моранді");
Buttons.Add("НСАВ за моделлю Нельсона");
break;
}

```

Після відновлення коду, був налаштований файл конфігурації C#-проекту, повний код файлу доступний в додатку Г.

3.4 Тестування ПЗ

Після створення програмного коду результатом розроблення стало програмне забезпечення для обчислення параметрів автоматизованих систем управління (рис. 3.3, рис. 3.4).

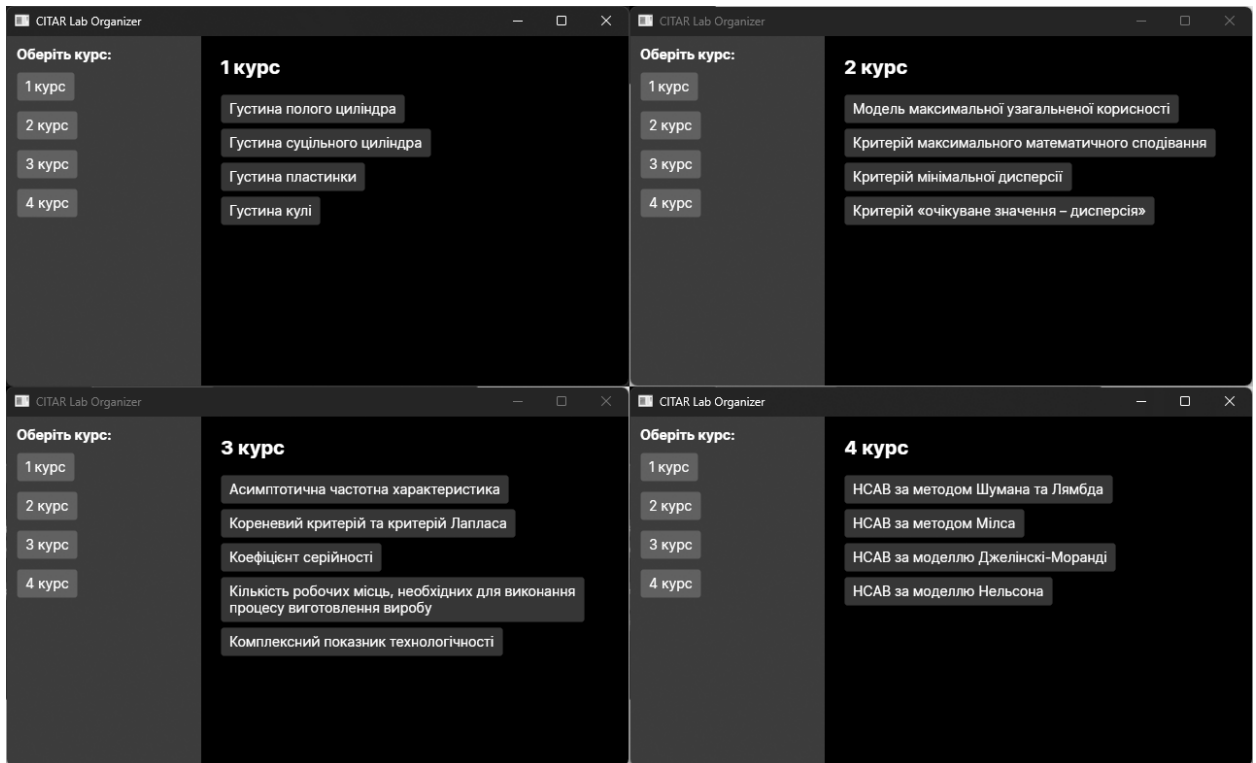


Рисунок 3.3 – Тестування інтерфейсу ПЗ

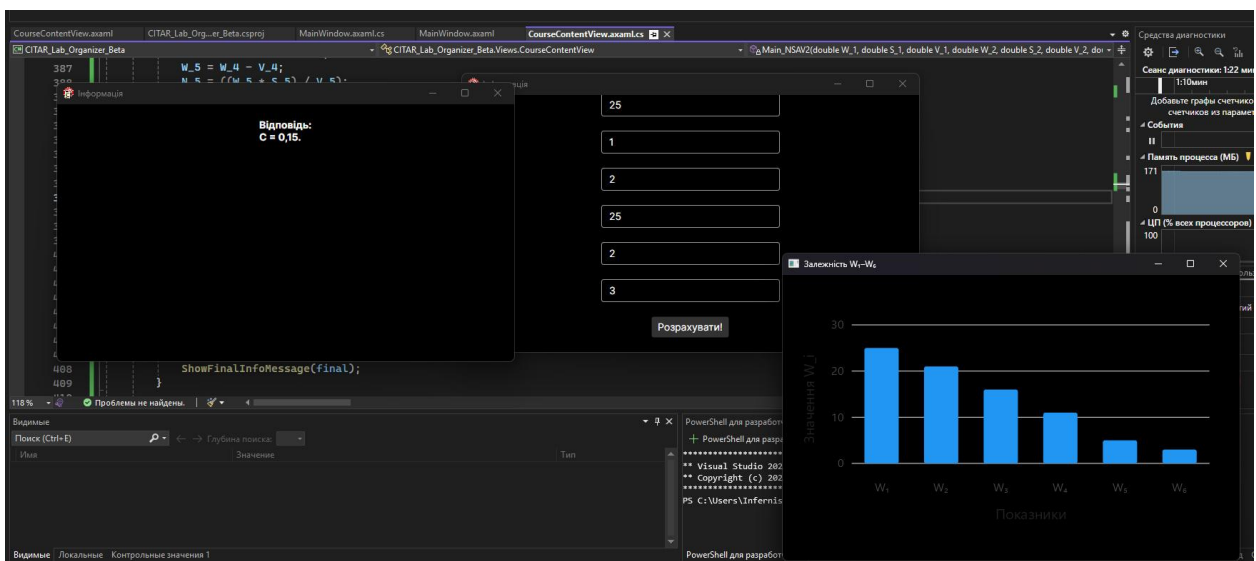


Рисунок 3.4 – Тестування інтерфейсу розрахунків

Для перевірки правильності розрахунків достатньо виконати будь-яку лабораторну або практичну роботу згідно з методичних вказівок. У якості прикладу такої роботи ідеально підійде третя практична робота з дисципліни «Проектування систем автоматизації».

Метою практичного заняття є набуття практичних навичок вирішення проблем моделювання надійності систем автоматизованого виробництва за допомогою моделі Желінські-Моранді. Необхідно визначити ймовірність відсутності п'ятої відмови. В результаті побудувати діаграму всіх відмов для кожного прогону ПЗ.

У якості прикладу робота була виконана за сьомим варіантом, дані для якого були вказані в методичних вказівках роботи. Дані для розрахунків наведені в таблиці 3.2.

Таблиця 3.2 – Дані для розрахунків

№варіанту	Первісну кількість помилок N	t_1	t_2	t_3	t_4
7	3	3	17	25	29

Модель Желінські-Моранді – модель надійності ПЗ систем і програмно-керованих засобів зв'язку (ПКЗЗ) – сукупність аналітичних виразів, що описують функціональні залежності показників надійності від часу і дозволяють отримати їх кількісну оцінку.

Модель заснована на припущеннях, що час до наступної відмови розподілено експоненціально, а інтенсивність відмов програми пропорційна кількості решти в програмі помилок.

Згідно з цими припущеннями ймовірність безвідмовної роботи ПЗ як функція часу t_i дорівнює

$$P(t_i) = e^{-\lambda \cdot t_i}, \quad (3.1)$$

де λ_i – інтенсивність відмов, яка пропорційна числу ще не виявлених i -тих помилок в програмі

$$\lambda_i = C_D (N - (i - 1)) = C_D (N - i + 1), \quad (3.2)$$

де C_D – коефіцієнт пропорційності;

N – початкова кількість помилок.

У (3.1) відлік часу починається від моменту останньої ($i - 1$)-ої відмови програми.

За методом максимуму правдоподібності на підставі (3.1), позначаючи через k номер прогнозованої відмови, отримаємо, що функція правдоподібності має вигляд

$$F = \prod_{i=1}^{k-1} C_D (N - i + 1) e^{-C_D (N - i + 1) t_i}. \quad (3.3)$$

Логарифмічна функція правдоподібності має вигляд

$$L = \ln F = \sum_{i=1}^{k-1} [\ln(C_D (N - i + 1)) - C_D (N - i + 1) t_i], \quad (3.4)$$

$$\frac{\partial L}{\partial C_D} = \sum_{i=1}^{k-1} \left[\frac{1}{C_D} - (N - i + 1) t_i \right] = 0. \quad (3.5)$$

$$\frac{\partial L}{\partial N} = \sum_{i=1}^{k-1} \left[\frac{1}{N - i + 1} - C_D t_i \right] = 0. \quad (3.6)$$

З (3.6) отримаємо

$$C_D = \frac{\sum_{i=1}^{k-1} \frac{1}{N-i+1}}{\sum_{i=1}^{k-1} t_i} \quad (3.7)$$

Підставивши (3.7) в (3.5) отримаємо:

$$(k-1) \cdot \frac{\sum_{i=1}^{k-1} t_i}{\sum_{i=1}^{k-1} \frac{1}{N-i+1}} = \sum_{i=1}^{k-1} (N-i+1)t_i \quad (3.8)$$

За відомих значень $k; t_1, t_2, \dots, t_k$ з (3.7) і (3.8) можна знайти значення параметрів моделі C_D і N , а потім інтенсивність відмов, час від останнього до наступної відмови t_{k+1} , ймовірність безвідмовної роботи через час t_{k+1} після останнього відмови.

Основною перевагою моделі є простота розрахунків.

Недолік цієї моделі полягає в тому, що при неточному визначенні величини N інтенсивність відмов програми може стати негативною, що призводить до неправильного результату.

Окрім того, передбачається, що при виправленні виявлених помилок не вносяться нові помилки, що теж не завжди виконується.

При $k = 5$ і $N = 3$, маємо:

$$\begin{aligned} & (k-1) \times \frac{\sum_{i=1}^{k-1} t_i}{\sum_{i=1}^{k-1} \frac{1}{N-i+1}} = \\ & = (k-1) \times \frac{t_1 + t_2 + t_3 + t_4}{\frac{1}{N-i_1+1} + \frac{1}{N-i_2+1} + \frac{1}{N-i_3+1} + \frac{1}{N-i_4+1}} = \\ & = (5-1) \times \frac{3 + 17 + 25 + 29}{\frac{1}{3-1+1} + \frac{1}{3-2+1} + \frac{1}{3-3+1} + \frac{1}{3-4+1}} = \end{aligned}$$

$$= 4 \times \frac{74}{\frac{1}{3} + \frac{1}{2} + 1} \approx 4 \times 40,43 \approx 162.$$

$$\begin{aligned} \sum_{i=1}^{k-1} (N - i + 1)t_i &= (N - 1 + 1)t_1 + (N - 2 + 1)t_2 + (N - 3 + 1)t_3 + \\ &+ (N - 4 + 1)t_4 = (3 - 1 + 1) \times 3 + (3 - 2 + 1) \times 17 + \\ &+ (3 - 3 + 1) \times 25 + (3 - 4 + 1) \times 29 = 9 + 34 + 25 = 68. \\ C_D &= \frac{\sum_{i=1}^{k-1} \frac{1}{N - i + 1}}{\sum_{i=1}^{k-1} t_i} = \frac{\frac{1}{N - i_1 + 1} + \frac{1}{N - i_2 + 1} + \frac{1}{N - i_3 + 1} + \frac{1}{N - i_4 + 1}}{t_1 + t_2 + t_3 + t_4} = \\ &= \frac{\frac{1}{N - i_1 + 1} + \frac{1}{N - i_2 + 1} + \frac{1}{N - i_3 + 1} + \frac{1}{N - i_4 + 1}}{t_1 + t_2 + t_3 + t_4} = \\ &= \frac{\frac{1}{3 - 1 + 1} + \frac{1}{3 - 2 + 1} + \frac{1}{3 - 3 + 1} + \frac{1}{3 - 4 + 1}}{3 + 17 + 25 + 29} \approx 0,025. \end{aligned}$$

$$\lambda_i = C_D(N - (i - 1)) = 0,025 \times (3 - (3 - 1)) = 0,025 \times 1 = 0,025.$$

Отже, середній час до наступної відмови становить:

$$t_3 = \frac{1}{\lambda_4} = \frac{1}{0,025} = 40 \text{ год.}$$

При $N = 4$, маємо:

$$\begin{aligned} (k - 1) \times \frac{\sum_{i=1}^{k-1} t_i}{\sum_{i=1}^{k-1} \frac{1}{N - i + 1}} &= \\ = (k - 1) \times \frac{t_1 + t_2 + t_3 + t_4}{\frac{1}{N - i_1 + 1} + \frac{1}{N - i_2 + 1} + \frac{1}{N - i_3 + 1} + \frac{1}{N - i_4 + 1}} &= \\ = (5 - 1) \times \frac{3 + 17 + 25 + 29}{\frac{1}{4 - 1 + 1} + \frac{1}{4 - 2 + 1} + \frac{1}{4 - 3 + 1} + \frac{1}{4 - 4 + 1}} &= \end{aligned}$$

$$= 4 \times \frac{74}{\frac{1}{4} + \frac{1}{3} + \frac{1}{2} + 1} \approx 3,8 \times 37 \approx 142.$$

$$\begin{aligned} \sum_{i=1}^{k-1} (N - i + 1)t_i &= (N - 1 + 1)t_1 + (N - 2 + 1)t_2 + (N - 3 + 1)t_3 + \\ &+ (N - 4 + 1)t_4 = (4 - 1 + 1) \times 3 + (4 - 2 + 1) \times 17 + \\ &+ (4 - 3 + 1) \times 25 + (4 - 4 + 1) \times 29 = 12 + 51 + 50 + 29 = 142. \\ C_D &= \frac{\sum_{i=1}^{k-1} \frac{1}{N - i + 1}}{\sum_{i=1}^{k-1} t_i} = \frac{\frac{1}{N - i_1 + 1} + \frac{1}{N - i_2 + 1} + \frac{1}{N - i_3 + 1} + \frac{1}{N - i_4 + 1}}{t_1 + t_2 + t_3 + t_4} = \\ &= \frac{\frac{1}{N - i_1 + 1} + \frac{1}{N - i_2 + 1} + \frac{1}{N - i_3 + 1} + \frac{1}{N - i_4 + 1}}{t_1 + t_2 + t_3 + t_4} = \\ &= \frac{\frac{1}{4 - 1 + 1} + \frac{1}{4 - 2 + 1} + \frac{1}{4 - 3 + 1} + \frac{1}{4 - 4 + 1}}{3 + 17 + 25 + 29} = \frac{\frac{1}{4} + \frac{1}{3} + \frac{1}{2} + 1}{74} \approx 0,028. \end{aligned}$$

$$\lambda_i = C_D(N - (i - 1)) = 0,027 \times (4 - (3 - 1)) = 0,028 \times 2 = 0,056.$$

Отже, середній час до наступної відмови становить:

$$t_3 = \frac{1}{\lambda_4} = \frac{1}{0,056} = \text{до 18 год.}$$

При $N = 5$, маємо:

$$\begin{aligned} (k - 1) \times \frac{\sum_{i=1}^{k-1} t_i}{\sum_{i=1}^{k-1} \frac{1}{N - i + 1}} &= \\ &= (k - 1) \times \frac{t_1 + t_2 + t_3 + t_4}{\frac{1}{N - i_1 + 1} + \frac{1}{N - i_2 + 1} + \frac{1}{N - i_3 + 1} + \frac{1}{N - i_4 + 1}} = \\ &= (5 - 1) \times \frac{3 + 17 + 25 + 29}{\frac{1}{5 - 1 + 1} + \frac{1}{5 - 2 + 1} + \frac{1}{5 - 3 + 1} + \frac{1}{5 - 4 + 1}} = \end{aligned}$$

$$= 4 \times \frac{74}{\frac{1}{5} + \frac{1}{4} + \frac{1}{3} + \frac{1}{2}} \approx 4 \times 57,8 = 231.$$

$$\begin{aligned} \sum_{i=1}^{k-1} (N - i + 1)t_i &= (N - 1 + 1)t_1 + (N - 2 + 1)t_2 + (N - 3 + 1)t_3 + \\ &+ (N - 4 + 1)t_4 = (5 - 1 + 1) \times 3 + (5 - 2 + 1) \times 17 + \\ &+ (5 - 3 + 1) \times 25 + (5 - 4 + 1) \times 29 = 216. \\ C_D &= \frac{\sum_{i=1}^{k-1} \frac{1}{N - i + 1}}{\sum_{i=1}^{k-1} t_i} = \frac{\frac{1}{N - i_1 + 1} + \frac{1}{N - i_2 + 1} + \frac{1}{N - i_3 + 1} + \frac{1}{N - i_4 + 1}}{t_1 + t_2 + t_3 + t_4} = \\ &= \frac{\frac{1}{N - i_1 + 1} + \frac{1}{N - i_2 + 1} + \frac{1}{N - i_3 + 1} + \frac{1}{N - i_4 + 1}}{t_1 + t_2 + t_3 + t_4} = \\ &= \frac{\frac{1}{5 - 1 + 1} + \frac{1}{5 - 2 + 1} + \frac{1}{5 - 3 + 1} + \frac{1}{5 - 4 + 1}}{3 + 17 + 25 + 29} = \frac{\frac{1}{5} + \frac{1}{4} + \frac{1}{3} + \frac{1}{2}}{74} \approx 0,017,3. \end{aligned}$$

$$\lambda_i = C_D(N - (i - 1)) = 0,017 \times (5 - (3 - 1)) = 0,017,3 \times 3 \approx 0,052.$$

Отже, середній час до наступної відмови становить:

$$t_3 = \frac{1}{\lambda_4} = \frac{1}{0,051} = 20 \text{ год.}$$

Для виконання цієї практичної роботи, потрібна дуже велика кількість розрахунків, по суті, однієї й тієї самої формули (3.8), просто з різними введеними значеннями. Для завдань із подібними формулами ідеально підійде програмна реалізація знаходження ПАТП, зі зручним інтерфейсом та полями введення. Рішення формули (3.8) можна реалізувати наступним програмним кодом:

```
void Main_NSAV3(double k, double N, double t1, double t2, double t3,
double t4, double i1, double i2, double i3, double i4, double i)
{
    double nm1p1, nm2p2, nm3p3, nm4p4;
```

```

nm1p1 = 1 / (N - i1 + 1);
nm2p2 = 1 / (N - i2 + 1);
nm3p3 = 1 / (N - i3 + 1);
nm4p4 = 1 / (N - i4 + 1);
if (i1 == N + 1) { nm1p1 = 0; }
if (i2 == N + 1) { nm2p2 = 0; }
if (i3 == N + 1) { nm3p3 = 0; }
if (i4 == N + 1) { nm4p4 = 0; }
double JM1, JM2, CD, lambdaI, t;
JM1 = ((k - 1) * ((t1 + t2 + t3 + t4) / (nm1p1 + nm2p2 + nm3p3 +
nm4p4)));
JM2 = (((N - i1 + 1) * t1) + ((N - i2 + 1) * t2) + ((N - i3 + 1) * t3) +
((N - i4 + 1) * t4));
CD = ((nm1p1 + nm2p2 + nm3p3 + nm4p4) / (t1 + t2 + t3 + t4));
lambdaI = (CD * (N - (i - 1)));
t = (1 / (lambdaI));
string final = ("JM1 = " + JM1 + "\nJM2 = " + JM2 + "\nCD = " + CD
+ "\nlambdaI = " + lambdaI + "\nt = " + t + ".");
ShowFinalInfoMessage(final);
}

```

У цьому кодї, значення k , N , t_1 , t_2 , t_3 , t_4 , i_1 , i_2 , i_3 , i_4 , i , CD , а також t_i та λ_i , подані як типи даних, а $JM1$ і $JM2$ являють собою значення формул (3.7) і (3.8). Значення k , N , t_1 , t_2 , t_3 , t_4 , i_1 , i_2 , i_3 , i_4 , t_i , λ_i , $JM1$, $JM2$ та CD вказані як дробові числа (double). Спочатку користувач вводить цілі числові значення k , N , t_1 , t_2 , t_3 , t_4 , i_1 , i_2 , i_3 , i_4 та i відповідно до його варіанту завдання, потім програма автоматично виконує розрахунки за формулами (3.7) і (3.8), та виводить значення за допомогою команди виведення. Перевіримо результати.

Результат розрахунків при $k = 5$ і $N = 3$ показаний на рисунку 3.5.

Інформація

НСАВ за моделлю Джелінскі-Моранді

5

3

3

17

25

29

Інформація

Відповідь:
JM1 = 161,45454545454547
JM2 = 68
CD = 0,024774774774774775
lambda4 = 0,024774774774774775
t = 40,36363636363636.

Рисунок 3.5 – Результат при $k = 5$ і $N = 3$

Отже, середній час до наступної відмови становить:

$$t_3 = \frac{1}{\lambda_4} = \frac{1}{0,025} = 40 \text{ год.}$$

Результат розрахунків при $N = 4$ показаний на рисунку 3.6.

Інформація

НСАВ за моделлю Джелінскі-Моранді

5

4

3

17

25

29

Інформація

Відповідь:
JM1 = 142,08
JM2 = 142
CD = 0,02815315315315315
lambdal = 0,0563063063063063
t = 17,76.

Рисунок 3.6 – Результат при $N = 4$

Отже, середній час до наступної відмови становить:

$$t_3 = \frac{1}{\lambda_4} = \frac{1}{0,056} = 18 \text{ год.}$$

Результат розрахунків при $N = 5$ показаний на рисунку 3.7.

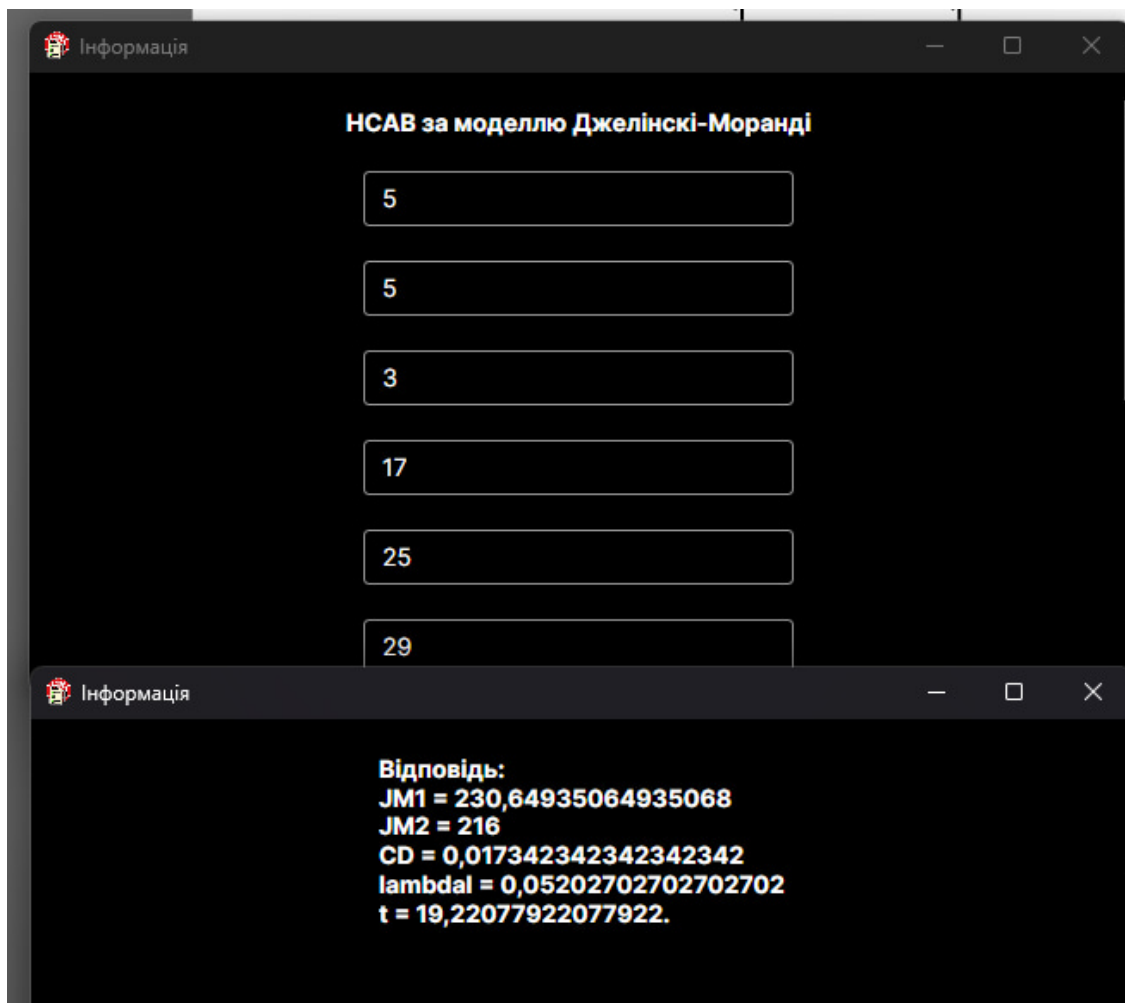


Рисунок 3.7 – Результат при $N = 5$

Отже, середній час до наступної відмови становить:

$$t_3 = \frac{1}{\lambda_4} = \frac{1}{0,051} = \text{до } 20 \text{ год.}$$

Використовуючи інформацію з навчального посібника з розробки кросплатформного програмного забезпечення на Avalonia [16], вдалося розробити ПЗ, яке працює абсолютно справно, враховуючи значення ПАТП навіть дещо точніше за розрахунки без її допомоги. Запропонований програмний код суттєво оптимізує процес моделювання надійності систем автоматизованого виробництва, зокрема на основі моделі Джелінські-Моранді, та значення ПАТП з дисципліни КІТВЗА [17-21], а також спрощує виконання аналогічних розрахункових задач. У зв'язку з цим виникає актуальна потреба

у створенні інструменту, який би дозволив студентам без зайвих зусиль підставляти необхідні значення у формули, отримувати результати без багаторазового повторення одних і тих самих дій, і при цьому забезпечував би можливість змінювати логіку обчислень відповідно до завдань. Важливою перевагою розробленого застосунку є також інтуїтивно зрозумілий інтерфейс, що дозволяє вводити кілька варіантів вхідних даних для одного й того ж алгоритму, не потребуючи повного перезапуску розрахунку.

Розроблена програма може бути корисною не лише студентам під час навчання, а й викладачам – для перевірки завдань, підготовки навчальних матеріалів та демонстрації принципів роботи розрахункових алгоритмів безпосередньо в аудиторії.

У підсумку, доцільність створення такого програмного забезпечення зумовлена реальною практикою навчального процесу. Його головна мета – зменшити обсяг рутинної роботи, щоб надати студентам більше часу для глибшого осмислення математичних моделей, принципів функціонування систем і суті інженерних розрахунків.

3.4 Розроблення та інтеграція логотипу ПЗ в програмний код

Було розроблено унікальний логотип – візуальний ідентифікатор ПЗ. Візуальний стиль логотипу поєднує у собі символіку навчального процесу, елементи графічного оформлення контрольних аркушів і стилізоване зображення ручки як інструменту запису та взаємодії з інформацією. Додатково, на фон було нанесено напівпрозорий логотип університету – Харківського національного університету радіоелектроніки (ХНУРЕ), що підкреслює належність розробки до навчального середовища. Логотип зображений на рисунку 3.8.



Рисунок 3.8 – Логотип ПЗ

Логотип був збережений у форматі ".ico" для зручної інтеграції в застосунок Avalonia UI як піктограма вікна. Для цього у проєкті було додано ресурс до каталогу "Assets", а в коді вікна використано властивість "Icon", що вказує шлях до логотипу:

```
var uri = new Uri("avares://CITAR_Lab_Organizer_Beta/Assets/logo-
final.ico");

var icon = new WindowIcon(AssetLoader.Open(uri));
var dialog = new Window
{
    Title = "Інформація",
    Icon = icon,
    Width = 640,
    Height = 360,
```

```

    MinWidth = 640,
    MinHeight = 360,
    MaxWidth = 640,
    MaxHeight = 360,
    WindowStartupLocation = WindowStartupLocation.CenterOwner,
    Content = scrollViewer
};
dialog.Show();
CopyButton_OnClick("Відповідь:\n" + title);

```

Також логотип відображається в меню вибору курсу поряд із заголовком застосунку, що створює більш професійне перше враження та забезпечує єдиний стиль оформлення інтерфейсу:

```

<Window xmlns="https://github.com/avaloniaui"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:vm="using:CITAR_Lab_Organizer_Beta.ViewModels"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-
compatibility/2006"
        mc:Ignorable="d" d:DesignWidth="640" d:DesignHeight="480"
        MinWidth="640"           MinHeight="480"           MaxWidth="640"
        MaxHeight="480"
        x:Class="CITAR_Lab_Organizer_Beta.Views.MainWindow"
        x:DataType="vm:MainWindowViewModel"
        Title="CITAR Lab Organizer"
        Icon="/Assets/logo-final.ico">
</Window>

```

Інтеграція графічного елемента не вплинула на продуктивність роботи програми, однак значно покращила її зовнішній вигляд та відповідність сучасним вимогам до користувачького інтерфейсу.

Результат наведений на рисунку 3.9.

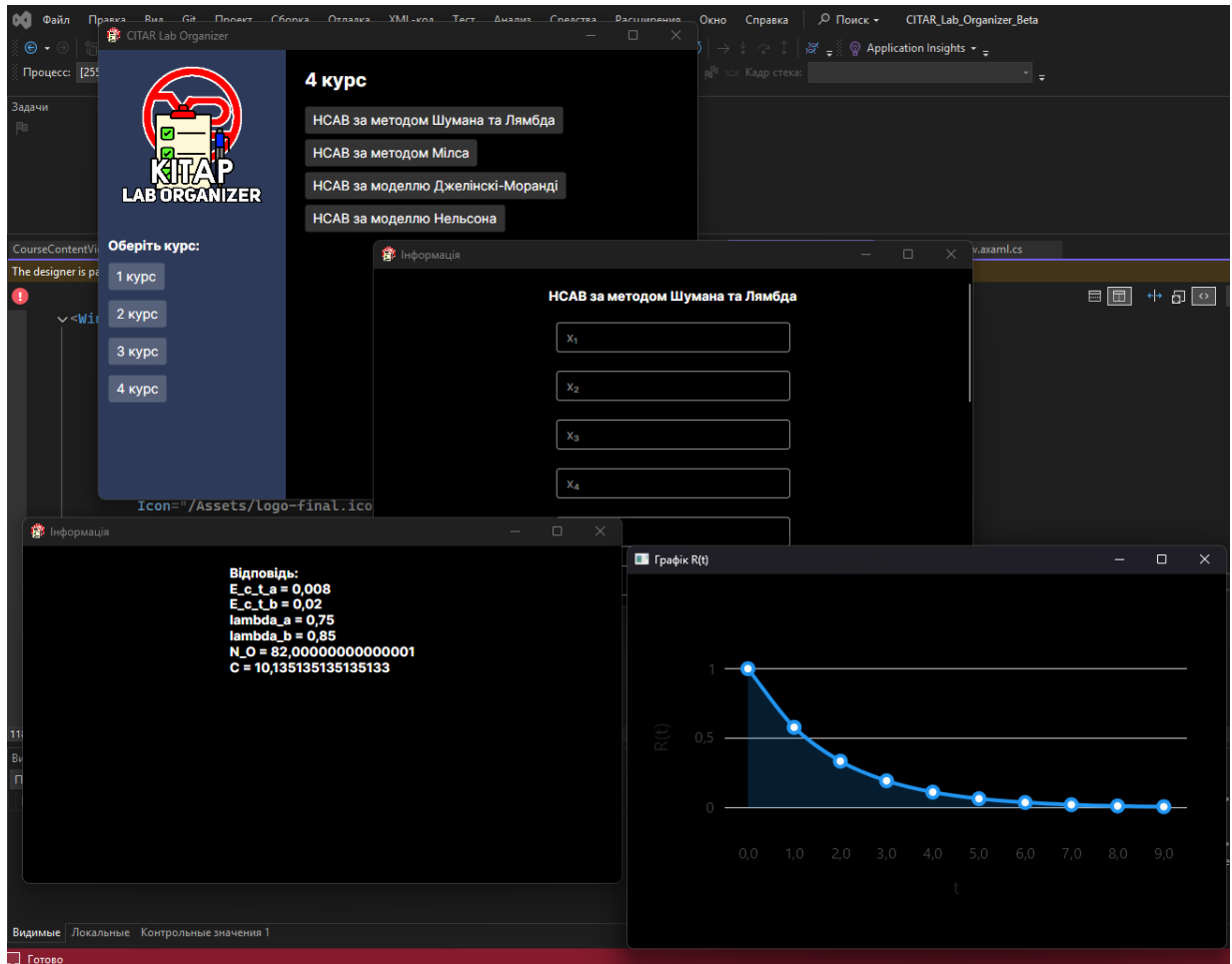
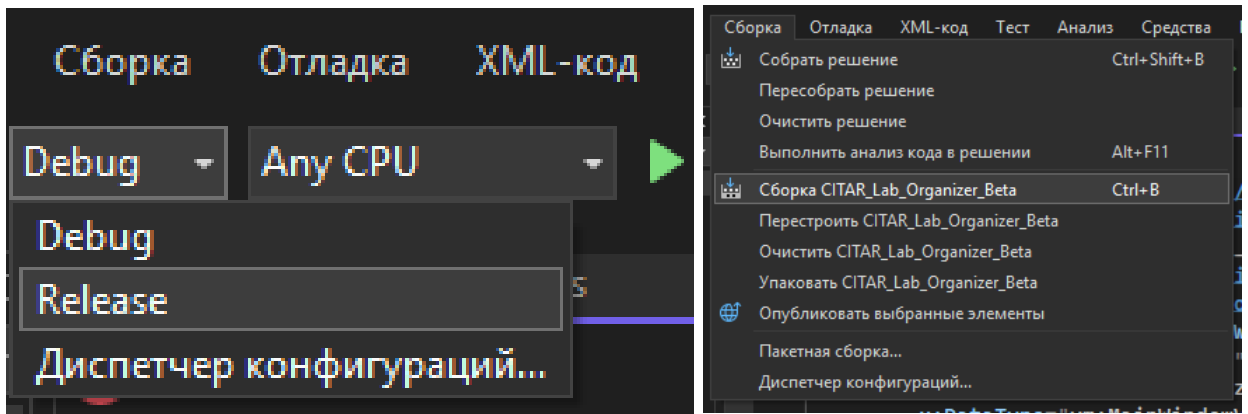


Рисунок 3.9 – Зміна інтерфейсу ПЗ з оновленням логотипом

Після інтеграції логотипу проведено повторне тестування ПЗ на наявність помилок, оптимізацію та стабільність. Виявлено, що дане ПЗ є повністю справним та виконує зазначені функції. Програмний засіб готовий до публікування.

3.5 Компіляція та публікація репозиторію на онлайн-платформу GitHub

Після розроблення ПЗ проведена зміна конфігурації на «Release» (рис. 3.10, а) та компіляція (рис. 3.10, б).



а)

б)

Рисунок 3.10 – Фінальні налаштування та компіляція програми

Після збірки ПЗ, репозиторій проекту був стиснутий до формату ".zip" та опублікований на сайті GitHub. Репозиторій доступний за посиланням [22].

3.6 Охорона праці

Робота над атестаційною роботою передбачала багато годин за екраном комп'ютера, зосередженість та інтелектуальне навантаження. Для зниження негативного впливу на здоров'я була організована відповідна система профілактичних заходів.

Насамперед було дотримано вимог щодо ергономічного облаштування робочого місця. Стіл мав відповідну висоту, що дозволяло зручно розмістити клавіатуру та монітор, а крісло забезпечувало підтримку попереку, що зменшувало навантаження на хребет. Відстань від очей до екрана становила приблизно 60–70 см, згідно з рекомендацій сучасних стандартів [23].

Особлива увага приділялася організації освітлення. Робота виконувалась у приміщенні з достатнім природним освітленням удень, а в темний час доби використовувались лампи з нейтральною колірною температурою, які мінімізують навантаження на очі.

Також було впроваджено режим роботи з регулярними перервами. Кожні 40–45 хв розробки передбачалась перерва тривалістю 15–20 хв,

що передбачено популярною технікою тайм-менеджменту, під назвою «Pomodoro Timer».

Окрім фізичних умов, було дотримано й принципів психологічного комфорту. Робочий процес організовувався так, щоб уникати надмірного емоційного виснаження: складні задачі чергувалися з простішими, а загальний час щоденної роботи не перевищував розумних меж. Було враховано й фактори шуму, температурного режиму та якості повітря в приміщенні.

Дотримання наведених рекомендацій дозволило уникнути ознак перевтоми, погіршення зору або порушення опорно-рухового апарату.

Таким чином, у рамках роботи над ПЗ було впроваджено комплекс заходів щодо охорони праці та безпеки життєдіяльності, які відповідають сучасним вимогам до організації інтелектуальної праці. Ці заходи сприяли підтриманню фізичного й психологічного здоров'я, забезпечили комфортні умови праці та високу якість виконання поставлених завдань.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи було розроблене програмне забезпечення, призначене для розрахунку параметрів автоматизованих технологічних процесів у межах лабораторного практикуму студентів. На першому етапі було проведено аналіз існуючих програмних рішень, що застосовуються у сфері навчання автоматизації, визначено їх функціональні особливості та виявлено недоліки, які було враховано при формуванні технічних вимог до майбутнього застосунку.

Було сформульовано загальну структуру інтерфейсу користувача, яка забезпечує інтуїтивне розташування елементів, зручний вибір навчального курсу, завдань і динамічне створення форм введення параметрів. Особливу увагу приділено механізму підтвердження вхідних даних, обробці винятків та відображенню результатів обчислень у зрозумілому вигляді.

У ході роботи було реалізовано функціонал вибору навчального курсу, відповідного переліку лабораторних і практичних робіт, а також окремі програмні методи для виконання кожного з розрахункових завдань. Було впроваджено механізм динамічного формування вікон вводу параметрів, реалізовано перевірку даних, обробку виключень і відображення результатів розрахунків у зручному вигляді.

У процесі розробки було створено функціональні модулі для реалізації широкого спектра математичних моделей, пов'язаних із визначенням надійності, корисності, інженерних критеріїв оптимізації та статистичних характеристик, що вивчаються у відповідних дисциплінах.

Для підвищення наочності результатів було впроваджено побудову графіків і діаграм на основі обчислених або введених даних, використовуючи сучасні бібліотеки візуалізації.

Було проведено тестування кожного з реалізованих методів, а також оцінено точність обчислень, стабільність роботи застосунку та зручність

взаємодії з користувачем. На основі отриманих результатів зроблено висновки щодо ефективності запропонованого рішення у навчальному процесі.

Особливу увагу приділено адаптації програми до специфіки методичних вказівок, а також забезпеченню необхідної точності та надійності виконуваних розрахунків.

Розроблене програмне забезпечення було завантажено на онлайн-платформу GitHub для можливості її завантаження та модифікування іншими користувачами.

Таким чином, було закріплено практичні навички з розробки програмного забезпечення для навчальних цілей, створення інтерактивного інтерфейсу, реалізації математичних моделей і візуалізації інженерних розрахунків.

Також матеріали з розробленого ПЗ були представлені на Міжнародній науково-технологічній конференції «Технологія-2025».

Подальшими кроками для модернізації розробки передбачаються: створення нової системи генерації покрокового рішення завдання, з демонстрацією кожного кроку підстановки чисел для ретельнішого інформування користувача стосовно того, які саме розрахунки проводяться для отримання результату.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. GitHub · Build and ship software on a single, collaborative platform. GitHub. URL: <https://github.com/> (дата звернення: 31.05.2025).
2. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. - 29 с.
3. Методичні вказівки з підготовки кваліфікаційної роботи для здобувачів першого (бакалаврського) рівня вищої освіти денної і заочної форми навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» / Упоряд.: І.Ш. Невлюдов, О.І. Филипенко, О.В. Токарева, С.П. Новоселов, О.В Сичова. Харків: ХНУРЕ, 2023. 64 с.
4. Навчальний посібник з підготовки кваліфікаційної роботи бакалавра для здобувачів вищої освіти денної і заочної форм навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології»: Навчальний посібник / І. Ш. Невлюдов, А.О. Андрусевич, О. В. Токарева, С. П. Новоселов, О. В. Сичова. – Харків : Видавництво Іванченка І. С., 2022. 151 с.
5. Automated teaching system using declarative problem solving approach : пат. CA2847769 Канада. № CA 2847769 ; заявл. 31.03.2014 ; опубл. 27.03.2015. 29 с. URL: <https://patents.google.com/patent/CA2847769A1> (дата звернення: 31.05.2025).
6. Automatic software production system : пат. US8365138B2 Сполучені Штати Америки (США) : G06F 9/44. № US12/284,947 ; заявл. 26.09.2008 ; опубл. 29.01.2013, Бюл. № В2. 77 с. URL:

<https://patents.google.com/patent/US8365138B2/en> (дата звернення: 31.05.2025).

7. Computer-aided education systems and methods: пат. US20040133532A1 Сполучені Штати Америки (США): G09B7/00, G09B7/02. № US10/640,385; заявл. 14.08.2003; опубл. 08.07.2004, Бюл. № 2004/0133532. 21 с. URL: <https://patents.google.com/patent/US20040133532A1/en> (дата звернення: 31.05.2025).

8. Croatia's Photomath raises \$23mn funding. Business news. URL: <https://www.intellinews.com/croatia-s-photomath-raises-23mn-funding-203348/?source=croatia> (дата звернення: 23.04.2025).

9. Методичні вказівки до лабораторних робіт з дисципліни «Фізика» (Частина 1, «Механіка та молекулярна фізика») для студентів усіх форм навчання напрямів 6.050201 «Системна інженерія», 6.050202 «Автоматизація та комп'ютерно-інтегровані технології», 6.050801 «Мікро- та наноелектроніка», 6.050802 «Електронні пристрої та системи», 6.050803 «Акустотехніка», 6.050901 «Радіотехніка», 6.050902 «Радіоелектронні апарати», 6.050903 «Телекомунікації», 6.051001 «Метрологія та інформаційно-вимірювальні технології», 6.051002 «Метрологія, стандартизація та сертифікація», 6.051004 «Опtotехніка», 6.170102 «Системи технічного захисту інформації», 6.170103 «Управління інформаційною безпекою» [Електронне видання] / Упоряд.: Коваленко О. М., Лазоренко О. В., Орел Р. П. – Харків: ХНУРЕ, 2014. – 83 с.

10. Методичні вказівки до лабораторних робіт з дисципліни «Системний аналіз складних систем управління» для студентів спеціальності 151 – Автоматизація та комп'ютерно-інтегровані технології [Текст] / Упоряд. І. В. Гребеннік, Л. В. Колесник, І. А. Урняєва, Р. В. Петрова. – Харків: ХНУРЕ, 2022. – 20 с.

11. Методичні вказівки до практичних занять з дисципліни «Теорія автоматичного управління» для студентів усіх форм навчання

спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» освітня програма Автоматизація та комп'ютерно-інтегровані технології [Електронне видання] / Упоряд.: О. В. Токарева. – Харків: ХНУРЕ, 2022. – 84 с.

12. Методичні вказівки до практичних занять з дисципліни «Комп'ютерно-інтегровані технології виробництва засобів автоматизації» для студентів усіх форм навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» / Упоряд.: І. В. Жарікова. – Харків: ХНУРЕ, 2022. – 44 с.

13. Методичні вказівки до практичних занять з дисципліни «Проектування систем автоматизації» / Сотник С. В., Харків: ХНУРЕ, 2022. – 31 с.

14. Avalonia UI – Open-Source .NET XAML Framework | WPF & MAUI Alternative. Avalonia UI – Open-Source .NET XAML Framework | WPF & MAUI Alternative. URL: <https://avaloniaui.net/> (дата звернення: 31.05.2025).

15. What is Windows Forms - Windows Forms. Microsoft Learn: Build skills that open doors in your career. URL: <https://learn.microsoft.com/uk-ua/dotnet/desktop/winforms/overview/> (дата звернення: 31.05.2025).

16. Основи розробки кросплатформного програмного забезпечення на Avalonia: Навчальний посібник / С. П. Новоселов, О. В. Сичова. Харків: Видавництво Іванченка І. С., 2024. 267 с. ISBN 978-617-8332-57-0. DOI: 10.30837/978-617-8332-57-0.

17. Невлюдов І. Ш. Комп'ютерно-інтегровані технології виробництва технічних засобів автоматизації. Частина 1: Підручник Харків: ФОП Панов А. М., 2021., 604 с. ISBN 978-617-7947-67-6.

18. Невлюдов І. Ш. Комп'ютерно-інтегровані технології виробництва технічних засобів автоматизації. Частина 2: Підручник

Кривий Ріг: видавець Чернявський Д. О., 2022. 424 с.з іл. ISBN 978-617-8045-44-9.

19. Igor Nevliudov, & et al. (2021). Automation of Mathematical Modeling of Physical and Technological Processes in the Electronic Devices Manufacture. Proceedings of the XII International Scientific Conference «Functional Basis of Nanoelectronics» – Odessa, September 20-24, 74-77.

20. Боцман І. В. Автоматизація процесу реконструювання параметрів технологічного процесу виготовлення гнучких структур // Всеукраїнська науково-практична Інтернет-конференція Автоматизація та комп'ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку. – Черкаси, 12-18 березня 2018 року. Черкаси. С. 75-77.

21. Igor Nevliudov, Iryna Botsman, Olena Chala, Kirill Khrustalev. Automated System Development for the Printed Circuit Boards Optical Inspection Using Machine Learning Methods // Proceedings of the 10-th International Scientific and Technical Conference «INFORMATION SYSTEMS AND TECHNOLOGIES (IST-2021)». – Odessa, September 13-19, 2021. – PP. 234-238.

22. GitHub - BohdanKyslyi/CITAR-Lab-Organizer-Beta: Програмне забезпечення для визначення параметрів автоматизованих технологічних процесів для лабораторного практикуму студентів. GitHub. URL: <https://github.com/BohdanKyslyi/CITAR-Lab-Organizer-Beta> (дата звернення: 31.05.2025).

23. Охорона праці: збірник задач: навч. посіб. / Б.В. Дзюндзюк, В.Г. Іванов [та ін.]; Харків. нац. ун-т радіоелектроніки. – Харків: ХНУРЕ, 2006. – 244 с.