

## АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ РЕВЕРСИВНОЇ ГЕНЕРАЦІЇ ТРИГОНОМЕТРИЧНИХ ЗАВДАНЬ ЗАСОБАМИ ОБ'ЄКТНО-ОРІЄНТОВАНОГО МОДЕЛЮВАННЯ

Світлана БОДНАРУК

Чернівецький національний університет  
імені Юрія Федьковича, Україна  
s.bodnaruk@chnu.edu.ua  
<https://orcid.org/0000-0002-4979-7669>

Назарій РИБЧИНСЬКИЙ ✉

Чернівецький національний університет  
імені Юрія Федьковича, Україна  
nazarii.rybchynskiy@gmail.com  
<https://orcid.org/0009-0001-4617-2823>

## ALGORITHMIC SUPPORT FOR REVERSE GENERATION OF TRIGONOMETRIC TASKS USING OBJECT-ORIENTED MODELING

Svitlana BODNARUK

Yuriy Fedkovych Chernivtsi National University, Ukraine  
s.bodnaruk@chnu.edu.ua  
<https://orcid.org/0000-0002-4979-7669>

Nazarii RYBCHYNSKYI ✉

Yuriy Fedkovych Chernivtsi National University, Ukraine  
nazarii.rybchynskiy@gmail.com  
<https://orcid.org/0009-0001-4617-2823>

### АНОТАЦІЯ

**Формулювання проблеми.** У статті представлено результати розробки та практичного впровадження алгоритмічної системи для автоматизації підготовки дидактичних матеріалів з розділу тригонометрії. Основна ідея дослідження ґрунтується на переході від традиційного механічного підбору задач до моделі реверсивного конструювання рівнянь. Запропонований підхід передбачає, що програмний алгоритм генерує математичну умову, виходячи із заздалегідь визначеної відповіді (коренів), що дає можливість вчителю миттєво створювати велику кількість варіативних завдань однакового рівня складності для забезпечення індивідуальної траєкторії навчання.

**Матеріали і методи.** Технічна реалізація системи виконана мовою програмування Python із залученням бібліотеки символічних обчислень SymPy. У роботі детально описано розроблену архітектуру класів, що охоплюють різні типи тригонометричних рівнянь: від базових до однорідних та таких, що розв'язуються методом введення нової змінної. Важливою технічною характеристикою алгоритму є автоматична верифікація області допустимих значень та фільтрація сторонніх коренів ще на етапі синтезу умови задачі, що гарантує абсолютну математичну точність кожного генерованого завдання.

**Результати.** Описана система є ефективною не лише для оперативного контролю знань, а й для організації самостійної роботи учнів у дистанційному та змішаному форматах. Наявність автоматично згенерованих покрокових розв'язків дозволяє здобувачам освіти самостійно опанувати алгоритми розв'язання складних завдань, мінімізуючи прогалини у знаннях без постійного втручання педагога. Ефективність запропонованої методики перевірено під час практичної апробації на базі ліцею у Чернівецькій області. Аналіз результатів експерименту показав, що систематичне використання персоналізованих завдань дозволило учням краще засвоїти методи тригонометричних перетворень, а середній бал успішності в експериментальних групах зріс на 15–20%.

**Висновки.** Авторами встановлено, що впровадження таких цифрових інструментів суттєво розвантажує вчителя, звільняючи час для творчої методичної діяльності та індивідуальної роботи з учнями. У висновках визначено перспективи масштабування реверсивного алгоритму для інших тем курсу алгебри та математичного аналізу.

### ABSTRACT

**Formulation of the problem.** The article presents the results of developing and implementing an algorithmic system to automate the preparation of trigonometry didactic materials. The core research idea is based on a transition from the traditional mechanical selection of problems to a model of reverse equation construction. The proposed approach implies that the software algorithm generates a mathematical condition based on a predetermined set of answers (roots), enabling the teacher to instantly create a vast number of variable tasks of the same complexity level to ensure a personalized learning trajectory.

**Materials and methods.** The system's technical implementation is implemented in Python using the SymPy symbolic computation library. The paper details the developed class architecture, covering various types of trigonometric equations, from basic ones to homogeneous equations and those solved by the variable-substitution method. A crucial technical characteristic of the algorithm is the automatic verification of the domain of permissible values and the filtering of extraneous roots during problem condition synthesis, which guarantees the absolute mathematical accuracy of each generated task.

**Results.** The described system is effective not only for rapid knowledge assessment but also for organizing students' independent work in distance and blended learning formats. The availability of automatically generated step-by-step solutions allows students to independently master algorithms for solving complex problems, minimizing knowledge gaps without constant teacher intervention. The effectiveness of the proposed methodology was verified during practical testing at a lyceum in the Chernivtsi region. Analysis of the experimental results showed that the systematic use of personalized tasks enabled students to better master trigonometric transformations, and the average academic performance in the experimental groups increased by 15–20%.

**Conclusions.** The authors demonstrate that implementing these digital tools significantly reduces teachers' workload, freeing up time for creative pedagogical activities and individualized work with students. The conclusions identify opportunities to scale the reverse algorithm to other topics in the algebra and mathematical analysis course.

**КЛЮЧОВІ СЛОВА:** тригонометричні рівняння; автоматизація навчання; реверсивне конструювання задач; мова Python; бібліотека SymPy; диференційоване навчання.

**KEYWORDS:** trigonometric equations; educational automation; reverse problem construction; Python language; SymPy library; differentiated instruction.

**ДЛЯ ЦИТУВАННЯ:** Боднарук С., Рибчинський Н. Алгоритмічне забезпечення реверсивної генерації тригонометричних завдань засобами об'єктно-орієнтованого моделювання. *Фізико-математична освіта*, 2026. Том 41. № 2. С. 18-27. <https://doi.org/10.31110/fmo2026.v41i2-02>.

**FOR CITATION:** Bodnaruk, S., & Rybchynskiy, N. (2026). Algorithmic support for reverse generation of trigonometric tasks using object-oriented modeling. *Physical and Mathematical Education*, 41(2), 18-27. <https://doi.org/10.31110/fmo2026.v41i2-02>.

## ВСТУП

**Постановка проблеми.** Сучасний освітній простір України переживає етап активної цифрової трансформації, що вимагає від вчителя математики нових підходів до організації навчального процесу. В умовах змішаного навчання особливо гостро постають питання забезпечення об'єктивного контролю знань та дотримання академічної доброчесності. Традиційні методи підготовки дидактичних матеріалів, що базуються на статичних підручниках, поступово втрачають свою ефективність через широку доступність цифрових збірок готового домашнього завдання, які нівелиують ідею самостійної роботи учня. Водночас ручне створення вчителем унікальних варіантів контрольних робіт для кожного учня є трудомістким процесом, що часто унеможливує реалізацію індивідуального підходу в умовах великої кількості учнів.

Актуальність теми дослідження зумовлена нагальною потребою розв'язання протиріччя між необхідністю забезпечення кожного учня унікальним навчальним завданням та обмеженими часовими ресурсами педагога. Використання сучасних засобів математичного моделювання та генеративних моделей дозволяє автоматизувати рутинні процеси створення завдань. Застосування мови програмування Python у поєднанні з бібліотеками символічних обчислень відкриває можливості для створення адаптивних завдань, що відповідають вимогам шкільної програми та враховують індивідуальні особливості учнів. Таким чином, розробка методики використання автоматизованих систем генерації завдань є своєчасною відповіддю на виклики сучасної освіти.

Наукова новизна одержаних результатів полягає в тому, що вперше обґрунтовано доцільність поєднання бібліотеки символічних обчислень SymPy з методичними вимогами української школи для автоматизації контролю знань. Удосконалені підходи до диференціації навчання через впровадження реверсивних алгоритмів генерації, які гарантують отримання методично доцільних («красивих») відповідей.

**Аналіз останніх досліджень і публікацій.** У низці сучасних досліджень обґрунтовується значний потенціал великих мовних моделей та інтелектуальних систем у генерації математичного контенту для STEM-дисциплін, що сприяє підвищенню якості освітньої діагностики та автоматизації створення навчальних матеріалів (Chan et al., 2025). Зокрема, розробка багатоагентних фреймворків для генерування математичних задач демонструє ефективність автоматизованого підходу, який забезпечує більш точне формулювання завдань і сприяє раціональному розподілу когнітивного навантаження між учасниками освітнього процесу (Karbasi et al., 2025).

Важливе місце в цьому контексті посідають системи символічних обчислень. Використання інструментів на базі бібліотеки SymPy для аналізу та автоматизованого розв'язування рівнянь розширює можливості дослідження структури математичних об'єктів і поглиблює концептуальне розуміння навчального матеріалу (Ali & Khusro, 2024). Окремі дослідження також засвідчують ефективність застосування чат-ботів, зокрема на основі ChatGPT, у процесі вивчення тригонометричних тотожностей у старшій школі, що позитивно впливає на розвиток логічного мислення та навчальної мотивації здобувачів освіти (Fitri et al., 2025).

Аналіз наведених праць дає підстави стверджувати, що регіональні акценти досліджень мають певні відмінності: зокрема, азіатські науковці активніше зосереджуються на використанні ChatGPT у шкільному курсі математики, тоді як європейські та американські автори частіше досліджують потенціал SymPy у викладанні вищої математики (математичного аналізу, лінійної алгебри) та в розробці відкритих систем автоматичного розв'язування рівнянь. Водночас застосування символічних обчислювальних систем у шкільному курсі математики наразі залишається недостатньо представленим у наукових публікаціях.

У цьому контексті особливої уваги потребує питання інтеграції зазначених підходів у систему навчання математики старшої школи України.

Організація навчання математики в сучасній українській старшій школі ґрунтується на принципах профільної диференціації, що передбачають чітке розмежування змісту освіти на рівні стандарту, на рівні профільного та поглибленого. Такий підхід зумовлює суттєві відмінності в методичних системах навчання, обсязі навчального часу та вимогах до математичної компетентності учнів. Змістова лінія «Тригонометричні рівняння та нерівності» є індикатором цієї диференціації, оскільки демонструє перехід від суто прикладного сприйняття математичних моделей (рівень стандарту) до глибокого теоретичного аналізу та дослідницької діяльності (профільний та поглиблений рівні). Аналіз нормативних документів, зокрема навчальних програм, затверджених Міністерством освіти і науки України (2017a,b,c), дозволяє виявити специфіку викладання цієї теми для кожної категорії здобувачів освіти.

Аналіз навчально-методичного забезпечення дозволяє простежити трансформацію програмних вимог у конкретні дидактичні матеріали. Розглянуті підручники демонструють різні авторські концепції викладання теми, що впливають на структуру завдань для розроблюваної інформаційної системи.

Підручник А. Г. Мерзляка («Алгебра і початки аналізу. 10 клас. Профільний рівень»), що також використовується для поглибленого вивчення, характеризується високим рівнем теоретичного узагальнення та математичної строгості (Мерзляк та ін., 2018).

У підручнику Є. П. Неліна (профільний рівень) реалізовано методику, спрямовану на формування узагальнених способів діяльності (Нелін, 2018).

Підручник О. С. Істера (профільний рівень) фокусується на чіткій алгоритмізації навчального процесу (Істер & Ергіна, 2018).

Проведений аналіз навчально-методичної літератури та нормативних документів дозволяє виявити ключову проблему: вчитель математики працює в умовах значної варіативності вимог та підходів. Якщо для рівня стандарту пріоритетним є засвоєння алгоритмів розв'язування найпростіших рівнянь (шаблонна діяльність), то профільний та поглиблений рівні вимагають від учнів гнучкості мислення, вміння комбінувати різні методи (заміна, розкладання, введення допоміжного кута), проводити дослідження ОДЗ та працювати з параметрами.

Сучасний ринок освітнього ПЗ пропонує інструменти, орієнтовані переважно на автоматичне розв'язування задач учнем, а не на генерацію дидактичних матеріалів учителем (Волянська & Лемішко, 2017). Системи символічної математики, такі як WolframAlpha, забезпечують високу точність та покрокові розв'язки, проте функціонують як інтелектуальні калькулятори, що вимагають ручного введення кожного рівняння. Мобільні застосунки типу Photomath базуються на технології OCR і є інструментами «споживання» готового контенту, що створює ризики для академічної доброчесності та не вирішує проблему створення нових завдань.

Інтерактивні середовища, зокрема GeoGebra, мають потужний потенціал для візуалізації та динамічного дослідження тригонометричних функцій, проте не пристосовані для швидкої підготовки статичних роздаткових матеріалів. Спеціалізовані сервіси на кшталт Symbolab пропонують режими тренування з автоматичною перевіркою відповідей, однак вони працюють у форматі онлайн-тренажерів для індивідуальних користувачів. Жоден із зазначених засобів не дозволяє вчителю оперативно згенерувати пакет унікальних варіантів контрольних робіт у форматі PDF із повною деталізацією розв'язків, що й обґрунтовує доцільність розробки запропонованої системи.

Отже, аналіз нормативної бази та методичного забезпечення підтвердив суттєву розбіжність у вимогах до вивчення тригонометрії залежно від профілю навчання. Складність опанування теми зумовлена високим рівнем абстракції та ризиком виникнення типових помилок (втрата коренів, поява сторонніх розв'язків), що потребує глибокої диференціації завдань.

Попри наявність потужних обчислювальних систем, на ринку освітнього ПЗ спостерігається дефіцит інструментів для автоматизованого створення варіативного контенту. Суперечність між необхідністю індивідуалізації навчання та обмеженістю існуючих сервісів (що працюють переважно як калькулятори) обґрунтовує актуальність розробки спеціалізованої системи для генерації тригонометричних рівнянь із гнучким налаштуванням складності.

**Мета дослідження:** Головною метою роботи є теоретичне обґрунтування, розробка та експериментальна перевірка методики використання автоматизованої системи генерації тригонометричних рівнянь для підвищення ефективності навчання алгебри в закладах загальної середньої освіти.

## МЕТОДИ ДОСЛІДЖЕННЯ

Дослідження базується на комплексному поєднанні аналізу науково-педагогічної літератури та методів алгоритмічного моделювання. Програмна реалізація системи здійснювалася засобами об'єктно-орієнтованого програмування (мова Python, бібліотека SymPy). Експериментальна перевірка запропонованої методики проводилася у першому семестрі 2025–2026 навчального року на базі ліцею у Чернівецькій області. У дослідженні взяли участь 29 учнів 10-го класу (з них 15 осіб увійшли до експериментальної групи та 14 – до контрольної). Для перевірки знань та вмінь було розроблено комплекс діагностичних завдань, що включав: розв'язання базових тригонометричних рівнянь, рівнянь, що зводяться до квадратних, та однорідних рівнянь першого і другого порядків. Ефективність підходу оцінювалася за результатами порівняльного аналізу академічної успішності контрольної та експериментальної груп. Основними критеріями оцінювання визначено якість засвоєння алгоритмів розв'язування тригонометричних рівнянь, а також рівень самостійності учнів, продемонстрований під час опанування матеріалу за допомогою автоматизованих покрокових розв'язків.

## РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

### 1 Методичні особливості та алгоритмізація розв'язання

Архітектура програмного продукту охоплює генерацію ключових класів тригонометричних рівнянь, реалізацію алгоритмів розв'язання яких враховує методичні особливості їх розв'язання та типові помилки учнів (Боднарук та ін., 2024). Зупинимось у статті коротко на деяких типах рівнянь.

#### 1.1 Рівняння, що зводяться до квадратних

У межах реалізованої моделі метод заміни змінної розглядається не лише як засіб зведення тригонометричного рівняння до алгебраїчного, а й як інструмент превентивного подолання типових помилок учнів. Алгоритм генерації та розв'язання враховує такі ключові аспекти:

- Автоматизація тотожних перетворень: програма базується на системі тригонометричних тотожностей, що дозволяють звести всі функції до єдиного аргументу. Це мінімізує помилки, пов'язані з неправильним застосуванням формул (зокрема, формул подвійного кута  $\cos 2x = 2\cos^2 x - 1 = 1 - 2\sin^2 x = \cos^2 x - \sin^2 x$ , які часто плутають учні).

- Контроль області значень: програмний модуль здійснює обов'язкову верифікацію коренів отриманого квадратного рівняння. Якщо  $t = \sin x$ ,  $t = \cos x$ , система автоматично акцентує увагу на умові  $|t| \leq 1$ . У покрокових розв'язках, що генеруються ботом, цей етап виділено як критичний, що привчає учнів до відсіювання сторонніх коренів на етапі повернення до початкової змінної.

- Коректність фінального етапу: оскільки помилки на етапі розв'язування найпростіших рівнянь  $\sin x = a$ ,  $\cos x = a$ ,  $\operatorname{tg} x = a$  є найбільш масовими, у програмі реалізовано чітку логіку виведення загальних формул коренів з урахуванням періодичності.

Така структура алгоритму дозволяє змістити фокус із механічних обчислень на концептуальне розуміння методу заміни, де кожен крок програми слугує методичною підказкою для усунення прогалин у знаннях.

### 1.2 Однорідні тригонометричні рівняння

Однорідними тригонометричними рівняннями  $n$ -го степеня називають рівняння, у яких усі доданки мають однаковий сумарний степінь щодо  $\sin x$  і  $\cos x$ :

$$a_0 \sin^n x + a_1 \sin^{n-1} x \cos x + \dots + a_n \cos^n x = 0,$$

де  $a_0, a_1, \dots, a_n$  – дійсні числа, не всі рівні нулю.

Специфічна структура цього рівняння дозволяє застосовувати універсальний алгоритм зведення до алгебраїчного рівняння шляхом ділення на  $\cos^n x$  або  $\sin^n x$  (залежно від структури), отримуючи рівняння щодо  $\operatorname{tg} x$  або  $\operatorname{ctg} x$ .

Методичні аспекти та автоматизація контролю. При розробці алгоритму генерації та розв'язання особливу увагу приділено декільком методичним викликам:

- Проблема надмірної перевірки ОДЗ. У шкільній практиці учні часто виконують формальну перевірку умови  $\cos x \neq 0$ , не розуміючи її логічного підґрунтя. Програмний модуль акцентує увагу на тому, що для однорідних рівнянь точки, де  $\cos x = 0$ , априорі не є коренями (оскільки  $\cos x$  та  $\sin x$  не набувають нульових значень одночасно). Це дозволяє алгоритму пропонувати лаконічні розв'язки без зайвих обчислень, розвиваючи в учнів критичне ставлення до стандартних процедур.

- Типові помилки розпізнавання. Найчастіше учні плутають однорідні рівняння з неоднорідними (наприклад,  $a \sin x + b \cos x = c$  при  $c \neq 0$ ). Програма генерує систему контрприкладів, що допомагає візуалізувати відмінність між однорідною структурою та рівняннями, які потребують використання формул подвійного аргументу або введення допоміжного кута.

- Помилки вибору степеня при діленні. У рівняннях вищих степенів учні нерідко ділять на  $\cos x$  замість на  $\cos^n x$ . Програмна реалізація покрокового розв'язку автоматично виділяє максимальний степінь як дільник, супроводжуючи це коментарем: "Максимальний степінь у рівнянні –  $n$ , тому ділимо на  $\cos^n x$ ". Для автоматизації процесу в системі реалізовано клас, який на етапі синтезу умови перевіряє сумарний степінь кожного монома. Якщо умова однорідності порушена (наприклад, через наявність вільного члена), алгоритм пропонує зведення до однорідного рівняння через основну тригонометричну тотожність ( $1 = \sin^2 x + \cos^2 x$ ), що є важливою методичною вправою для учнів профільних класів.

### 1.3 Рівняння, що містять обернені тригонометричні функції

Робота з рівняннями, що містять обернені тригонометричні функції, передбачає відмову від шаблонних перетворень на користь глибокого аналізу областей визначення та значень. Оскільки композиції прямих та обернених функцій (наприклад,  $\sin(\arcsin x) = x$ ) є тотожними лише на обмежених проміжках, програмний алгоритм включає обов'язковий етап верифікації умов існування розв'язків, що мінімізує ризик появи сторонніх коренів у підсумкових дидактичних матеріалах. На відміну від учнів, які часто ігнорують обмеження  $|x| \leq 1$ , бібліотека SymPy дозволяє автоматизувати перевірку цих умов ще на етапі ініціалізації об'єктів класу рівнянь.

## 2 Розробка програмного забезпечення для генерації завдань

### 2.1 Обґрунтування вибору технології розробки

Вибір інструментарію для реалізації програмної системи генерації тригонометричних рівнянь базувався на аналізі специфіки предметної області, яка вимагає високої точності математичних обчислень, гнучкості у маніпулюванні символічними виразами та можливості динамічного розширення функціоналу. Ключовим завданням розробки є створення адаптивної системи, здатної не лише генерувати математичні задачі, але й формувати покрокові пояснення їх розв'язання, що накладає суттєві обмеження на вибір мови програмування та бібліотек.

Основною мовою реалізації було обрано Python. Даний вибір обумовлений наявністю розвиненої екосистеми для наукових та математичних обчислень, а також високим рівнем абстракції, що дозволяє зосередитися на алгоритмічній складовій задачі, мінімізуючи витрати часу на вирішення низькорівневих питань управління пам'яттю. Інтерпретована природа мови та динамічна типізація забезпечують можливість швидкого прототипування складних алгоритмів генерації, що є критично важливим на етапі підбору параметрів для отримання «табличних» значень тригонометричних функцій. Крім того, синтаксична лаконічність Python сприяє підтримці чистоти коду та його читабельності, що є важливим фактором при реалізації складної бізнес-логіки математичних перетворень.

Центральним компонентом системи є бібліотека символічної математики SymPy. Використання чисельних методів (наприклад, бібліотеки NumPy) у даному контексті є недоцільним, оскільки освітні задачі вимагають точних аналітичних розв'язків (у вигляді дробів, коренів, чисел  $\pi$ ), а не їх наближених десяткових значень. SymPy дозволяє оперувати математичними об'єктами у символічному вигляді, що дає змогу виконувати такі операції, як спрощення виразів, розкриття дужок, знаходження коренів рівнянь у загальному вигляді та перевірку тотожностей. Саме завдяки функціоналу цієї бібліотеки реалізовано механізми перевірки властивостей коефіцієнтів (цілочисельність, ірраціональність), що необхідно для фільтрації некоректних або занадто складних для шкільного курсу варіантів рівнянь.

Архітектура програмного комплексу побудована на принципах об'єктно-орієнтованого програмування (ООП), що дозволило створити модульну та розширювану систему. Було використано патерн проектування «Шаблонний метод» (Template Method), реалізований через базовий абстрактний клас. Цей клас визначає загальний скелет алгоритму роботи з рівняннями: ініціалізацію, генерацію параметрів, розв'язання та формування покрокового пояснення, тоді як конкретні підкласи реалізують специфічну логіку для кожного типу тригонометричних рівнянь (наприклад, однорідних, симетричних чи таких, що розв'язуються методом заміни). Такий підхід забезпечує дотримання принципу відкритості/закритості (Open/Closed Principle), дозволяючи додавати нові типи рівнянь без модифікації існуючого коду базової логіки. Поліморфізм дозволяє системі уніфіковано обробляти різні типи рівнянь, викликаючи відповідні методи генерації та розв'язання залежно від контексту.

Особливу увагу приділено алгоритмічному підходу до генерації задач. Замість прямого методу, який полягає у випадковому виборі коефіцієнтів з подальшою перевіркою дискримінанта, було застосовано метод «зворотного інжинірингу». Суть методу полягає у першочерговому виборі «гарних» (табличних) коренів або проміжних значень змінної, на основі яких за допомогою теореми Вієта або обернених тригонометричних перетворень обчислюються необхідні коефіцієнти рівняння. Це гарантує, що згенеровані задачі завжди матимуть розв'язок у межах шкільної програми та будуть методично коректними. Реалізація цього підходу вимагала використання можливостей Python для роботи з раціональними числами та ірраціональностями, що також забезпечується бібліотекою SymPy.

Для відображення математичних формул у зрозумілому для користувача вигляді використано мову розмітки LaTeX. Вона є стандартом де-факто у науковій спільноті для типографіки складних математичних виразів. Інтеграція SymPy дозволяє автоматично конвертувати внутрішні об'єкти програми (символьні вирази) у LaTeX-рядки. Це забезпечує високу якість візуалізації умов задач та кроків розв'язання, дозволяючи коректно відображати дроби, степені, грецькі літери та математичні оператори. У поєднанні з шаблонними рядками (f-strings) мови Python, це дозволило побудувати гнучку систему генерації текстових пояснень, де динамічно підставляються обчислені значення та формули.

Таким чином, комплексне використання мови Python, бібліотеки символьних обчислень SymPy, об'єктно-орієнтованої архітектури та мови розмітки LaTeX створює надійний фундамент для функціонування системи. Такий технологічний стек забезпечує необхідну точність математичних операцій, гнучкість у налаштуванні складності завдань та високу якість вихідних матеріалів.

## 2.2 Архітектура розробленої системи

Проектування архітектури інформаційної системи генерації навчального математичного контенту здійснювалося з дотриманням принципів системного аналізу та об'єктно-орієнтованого підходу. Головною метою при розробці архітектурного рішення було створення гнучкого, модульного середовища, яке дозволяє легко інтегрувати нові типи математичних задач без необхідності внесення змін у наявне ядро системи. Архітектура програмного засобу може бути класифікована як багаторівнева, де чітко розмежовані шари бізнес-логіки (математичне моделювання), управління даними (символьні обчислення) та представлення (генерація LaTeX-документів).

Центральним елементом архітектури є ієрархічна структура класів, побудована на базі абстрактного класу `TrigonometricEquation`. Цей клас виступає фундаментальним будівельним блоком системи, визначаючи уніфікований інтерфейс для всіх типів тригонометричних рівнянь. У ньому інкапсульовано загальні механізми, необхідні для функціонування будь-якого математичного об'єкта системи: ініціалізацію символьних змінних бібліотеки SymPy, управління життєвим циклом об'єкта рівняння, а також методи експорту результатів у формат PDF.

Для забезпечення розширюваності системи було застосовано поведінковий патерн проектування «Шаблонний метод» (Template Method). У базовому класі реалізовано каркас основного алгоритму обробки рівняння в конструкторі `__init__`, який послідовно викликає методи генерації умови, розв'язання та формування пояснень. При цьому самі методи `__generate()`, `__solve()` та `__build_solution_steps()` оголошені як абстрактні або містять базову реалізацію, що підлягає перевизначенню в дочірніх класах. Такий підхід гарантує, що додавання нового типу рівняння (наприклад, рівняння, що розв'язується введенням допоміжного кута) вимагає лише створення нового класу-спадкоємця та реалізації специфічної для нього математичної логіки, залишаючи інфраструктурний код незмінним. Це забезпечує виконання принципу відкритості/закритості (Open/Closed Principle) SOLID (Рис. 1).

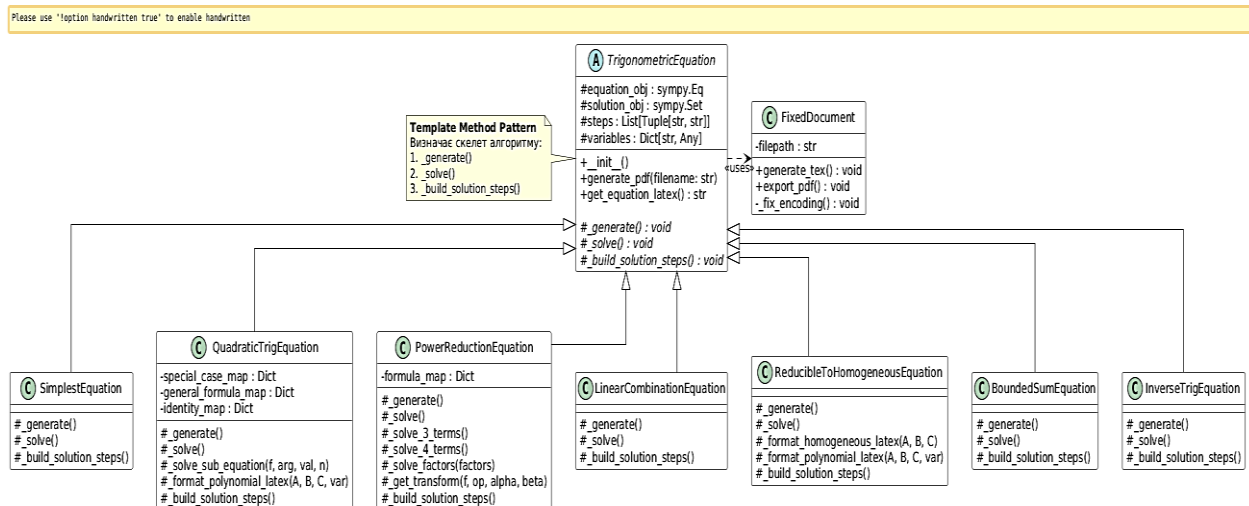


Рис. 1. Часткова діаграма класів розробленої системи

Джерело: авторська розробка

Спеціалізовані класи, такі як `HomogeneousEquation` (однорідні рівняння) або `InverseTrigEquation` (рівняння з оберненими тригонометричними функціями), виступають як контейнери для специфічних алгоритмів. Наприклад, клас `QuadraticTrigEquation` містить логіку зведення тригонометричного рівняння до алгебраїчного квадратного рівняння шляхом заміни змінної, включаючи перевірку дискримінанта та фільтрацію коренів, що не входять до області визначення. Така інкапсуляція дозволяє ізолювати складність окремих математичних методів, роблячи загальну структуру системи прозорою та керованою. При цьому алгоритм проходить через вузол перевірки типу рівняння, де для однорідних структур застосовується оптимізований шлях без надлишкових обчислень ОДЗ.

Важливою особливістю архітектури є підхід до управління даними. Система відмовляється від використання примітивних типів даних (чисел з плаваючою комою) на користь символічних об'єктів бібліотеки SymPy. Це дозволяє уникнути помилок округлення, властивих машинній арифметиці, та зберігати математичні константи у їх точному аналітичному вигляді протягом усього циклу обробки. Внутрішній стан кожного об'єкта рівняння зберігається у спеціалізованому словнику атрибутів self.variables. Цей словник виконує роль контекстного сховища, куди під час генерації записуються згенеровані коефіцієнти, проміжні змінні та корені.

Використання єдиного контекстного сховища є критично важливим для етапу формування методичних пояснень. При генерації покрокового розв'язання система звертається до цього словника для динамічного заповнення текстових шаблонів. Це дозволяє створювати адаптивні пояснення, де текст змінюється залежно від значень коефіцієнтів (наприклад, вибір формули коренів для парного або непарного коефіцієнта, або зміна знаку в формулі залежно від чверті одиничного кола).

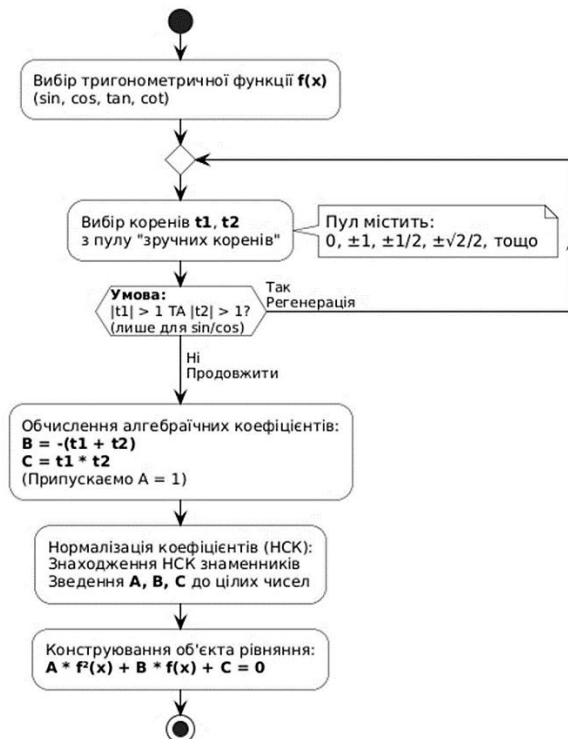
**2.3 Алгоритм генерації та перевірки коректності тригонометричних рівнянь**

Основним викликом при автоматизованій генерації математичних задач навчального характеру є забезпечення дидактичної цінності отриманого матеріалу. Прямий метод генерації, що полягає у випадковому виборі числових значень коефіцієнтів рівняння, у контексті тригонометрії є неефективним. Це зумовлено тим, що довільний набір коефіцієнтів з високою ймовірністю призводить до появи ірраціональних дискримінантів, нетабличних значень обернених тригонометричних функцій або відсутності дійсних коренів взагалі. Для вирішення цієї проблеми у розробленій системі застосовано стратегію конструктивної генерації, відому як метод «зворотного інжинірингу» або метод генерації «від відповіді».

Для класу рівнянь, що зводяться до алгебраїчних (квадратних відносно однієї тригонометричної функції, однорідних тощо), алгоритм ініціюється вибором множини допустимих значень заміни змінної. У системі сформовано попередньо визначені пули дидактично доцільних значень. До цього набору входять цілі числа, раціональні дроби та ірраціональні числа, що відповідають табличним значенням тригонометричних функцій.

Блок-схема алгоритму генерації для таких рівнянь представлена на рисунку 2 та складається з наступних етапів:

1. Стохастичний вибір коренів. З пулу допустимих значень випадковим чином обирається пара коренів. На цьому етапі застосовуються фільтри сумісності: наприклад, для рівнянь відносно синуса та косинуса відкидаються пари, де обидва корені за модулем перевищують одиницю, оскільки це призведе до відсутності розв'язків.
2. Побудова алгебраїчного ядра. На основі теореми Вієта формується зведене квадратне рівняння.
3. Нормалізація коефіцієнтів. Оскільки отримані коефіцієнти можуть бути дробовими або містити ірраціональність у знаменниках, виконується процедура приведення до цілочисельного або стандартного ірраціонального вигляду. Система обчислює найменше спільне кратне знаменників усіх коефіцієнтів та домножує на нього все рівняння. Це дозволяє отримати рівняння вигляду, що є звичним для шкільних підручників.
4. Трансформація у тригонометричний вигляд. Змінна замінюється на відповідну тригонометричну функцію.



**Рис. 2. Блок-схема алгоритму генерації квадратних тригонометричних рівнянь**

*Джерело: авторська розробка*

Для рівнянь виду (LinearCombinationEquation) застосовується інший підхід, оскільки прямий вибір часто призводить до «некрасивого» значення допоміжного кута, яке неможливо обчислити без калькулятора.

Алгоритм базується на фіксації цільового допоміжного кута. Процес генерації відбувається у такій послідовності: Обирається бажаний кут зсуву з множини табличних кутів, наприклад.

Обирається бажана амплітуда – гіпотенуза трикутника коефіцієнтів.

Коефіцієнти розраховуються як катети прямокутного трикутника.

Вільний член обирається як добуток амплітуди на табличне значення синуса або косинуса. Такий підхід гарантує, що при діленні рівняння учні отримають впізнавані значення тригонометричних функцій, що дозволить згорнути вираз у формулу синуса або косинуса суми.

Важливою складовою алгоритмічного забезпечення є багаторівнева система перевірки коректності (валідації) згенерованих задач. Перевірка здійснюється як на етапі генерації параметрів, так і на етапі формування кінцевого об'єкта рівняння.

Перший рівень валідації – перевірка області визначення та множини значень. Для рівнянь система автоматично відстежує, щоб хоча б один із коренів алгебраїчного рівняння-аналога належав відрізку  $[-1; 1]$ . Якщо ця умова не виконується, генерується виключна ситуація, яка перехоплюється керуючим алгоритмом, і процес генерації перезапускається з новими параметрами. Для рівнянь з тангенсом та котангенсом перевіряється, щоб аргумент функції не потрапляв у точки розриву.

Другий рівень валідації – перевірка на виродженість та тривіальність. Алгоритми містять евристичні правила, що запобігають створенню рівнянь, де старший коефіцієнт дорівнює нулю (перетворення квадратного рівняння у лінійне), або де всі коефіцієнти скорочуються до тотожності. Зокрема, у класі `PowerReductionEquation` (рівняння на пониження степеня) реалізовано перевірку на унікальність аргументів функцій, щоб уникнути ситуації взаємного знищення доданків при алгебраїчних перетвореннях (Рис. 3).

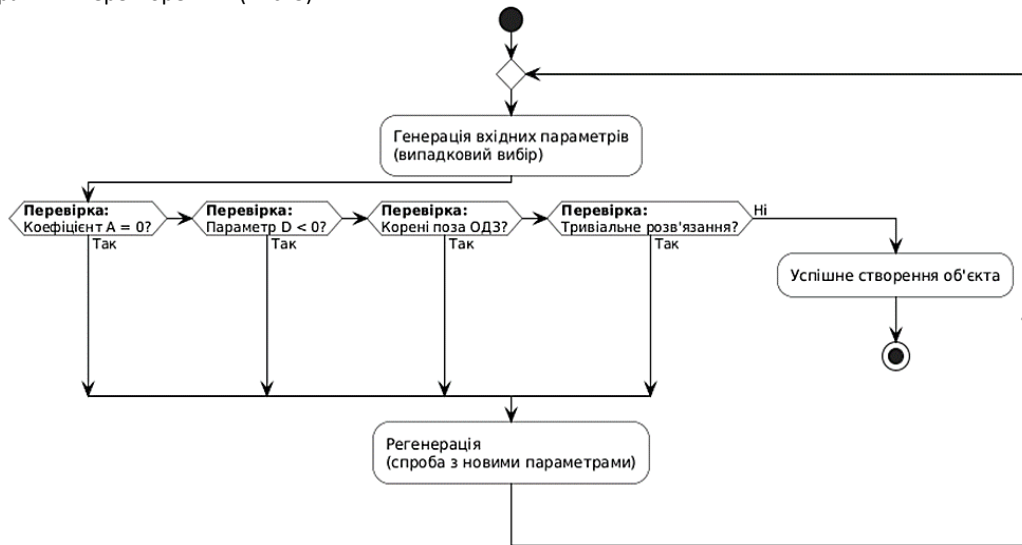


Рис. 3. Схема валідації параметрів

Джерело: авторська розробка

Третій рівень – перевірка структурної цілісності. Завдяки використанню бібліотеки символічної математики `SymPy`, система виконує спрощення згенерованих виразів `sympy.simplify()`. Це дозволяє виявити та усунути надлишкову складність (наприклад, скоротити дроби в коефіцієнтах або прибрати зайві мінуси) ще до того, як задача буде представлена користувачеві.

Запропоновані алгоритми забезпечують стабільну генерацію математично коректних, різноманітних та методично виважених тригонометричних рівнянь, що є необхідною умовою для використання розробленої системи в освітньому процесі.

Таким чином, спроектовано та програмно реалізовано систему автоматизованої генерації тригонометричних рівнянь, що базується на інтеграції мови `Python`, бібліотеки символічних обчислень `SymPy` та системи комп'ютерної верстки `LaTeX`. Обґрунтований вибір технологічного стека дозволив забезпечити абсолютну аналітичну точність операцій, уникнувши похибок округлення, властивих чисельним методам, та гарантувати високу поліграфічну якість візуалізації математичних формул у вихідних документах.

Архітектура комплексу, побудована на принципах об'єктно-орієнтованого програмування із застосуванням патерна «Шаблонний метод», забезпечила модульність системи та уніфікацію процесів обробки різноманітних рівнянь. Ключовим досягненням роботи стала реалізація алгоритмів конструктивної генерації методом «зворотного інжинірингу», які, у поєднанні з багаторівневою системою валідації параметрів, гарантують створення методично коректних завдань із прогнозованими «зручними» розв'язками, що робить розроблену систему ефективним інструментом для створення дидактичного контенту.

### 3 Типологія та класифікація згенерованих завдань

Розроблена система покриває повний спектр тригонометричних рівнянь, передбачених шкільною програмою поглибленого рівня та університетськими курсами елементарної математики. Реалізовані алгоритми («генератори») можна класифікувати за математичним методом розв'язання, що лежить в їх основі.

Фундаментом системи є генерація рівнянь, що зводяться до алгебраїчних шляхом заміни змінної. Сюди входять:

- найпростіші рівняння (`SimplestEquation`): базові конструкції, які є будівельними блоками для всіх інших типів;

- квадратні рівняння (QuadraticTrigEquation): рівняння виду  $A f^2(x) + B f(x) + C = 0$ , що розв'язуються заміною  $t = f(x)$ ;
  - рівняння з формулами подвійного аргументу (DoubleAngleToQuadraticEquation): задачі, що потребують попереднього спрощення для зведення до квадратного.
- Іншою групою алгоритмів є методи розкладання на множники та перетворень, дана група формул рівняння, розв'язання яких вимагає застосування тригонометричних тотожностей для факторизації виразу:
- сума/різниця в добуток (SumToProductEquation): використання формул  $\sin(\alpha) \pm \sin(\beta)$  для отримання добутку, що дорівнює нулю;
  - метод групування (GroupingEquation): складніші вирази з 4-ма і більше доданками, які потрібно згрупувати для винесення спільного множника;
  - пониження степеня (PowerReductionEquation): рівняння з квадратами синуса чи косинуса, де ключовим кроком є перехід до функцій подвійного аргументу;
  - структурні методи (однорідність та лінійна комбінація) є окремою категорією алгоритмів, що відповідає за генерацію рівнянь зі специфічною структурою;
  - однорідні рівняння (HomogeneousEquation) та ті, що зводяться до них (ReducibleToHomogeneousEquation): рівняння, де всі члени мають однаковий сумарний степінь, що дозволяє виконати ділення на  $\cos^n(x)$  і перейти до  $tg(x)$ ;
  - метод допоміжного кута (LinearCombinationEquation): рівняння виду  $A \sin(x) + B \cos(x) = C$ . Система гарантує підбір коефіцієнтів  $a, b$ , що відповідають табличним значенням синуса і косинуса.
- Для перевірки поглиблених знань реалізовано генератори, що базуються на неочевидних підстановках:
- симетричні рівняння (SymmetricEquation): використовують заміну  $t = \sin(x) \pm \cos(x)$ , що дозволяє виразити добуток  $\sin(x) \cos(x)$  через  $t$ ;
  - універсальні та спеціальні заміни (TanSubstitutionEquation, SumTanCotanEquation): рівняння, що розв'язуються через  $t = tg(x)$  або  $t = tg(x) + ctg(x)$ .
- Також реалізовано алгоритм генерації для методу оцінки (BoundedSumEquation): нестандартні рівняння (наприклад,  $\sin(x) + \cos(5x) = 2$ ), що розв'язуються через аналіз області значень.
- Останнім алгоритмом є генерація рівнянь з арк-функціями (InverseTrigEquation): алгебраїчні рівняння, де змінна знаходиться під знаком оберненої тригонометричної функції.
- Результатом роботи системи є файл у форматі PDF, зверстаний за допомогою системи LaTeX. Такий формат є стандартом для науково-технічної документації, забезпечуючи ідеальну читабельність формул, крос-платформність та готовність до друку.

#### 4 Аналіз якості та методологічні зауваження

Використання бібліотеки SymPy гарантує математичну точність, проте апробація системи виявила потребу в адаптації машинних результатів до шкільних стандартів. Ключовою проблемою є специфіка канонізації розв'язків: SymPy часто генерує надлишкові об'єднання множин (наприклад, окремі серії коренів замість однієї компактною формули). Для подолання цього ефекту в програмному комплексі реалізовано евристичні алгоритми постобробки, які виконують примусове «згортання» періодичних серій у загальноприйнятий формат.

Іншим аспектом є особливості автоматичного спрощення виразів. Критерії мінімізації символів у SymPy іноді призводять до «непедагогічних» трансформацій, що ускладнюють розуміння логіки перетворень (наприклад, передчасне розкриття дужок або нестандартна робота з ірраціональностями). Інтегрований модуль примусового форматування поліномів блокує стандартні спрощення на етапі візуалізації, забезпечуючи звичний для учнів вигляд коефіцієнтів.

Також враховано архітектурну детермінованість методів розв'язання. Попри існуючу в математиці варіативність підходів (наприклад, вибір між методом допоміжного кута та діленням у однорідних рівняннях), система безальтернативно обирає алгоритм, заданий параметрами генерації. Таке обмеження є свідомим методичним рішенням, орієнтованим на відпрацювання конкретної навчальної теми.

#### 5 Оцінка надійності та верифікація результатів

Розроблена система базується на ядрі символічних обчислень SymPy, що гарантує фундаментальну математичну коректність логіки. Разом з тим, складність алгоритмізації тригонометричних перетворень зумовлює виникнення поодиноких технічних аномалій у граничних випадках генерації параметрів (менше 1% випадків за результатами апробації).

Для нівелювання цих ризиків у програмі реалізовано:

- Механізм перехоплення виключень (Exception Handling): при виявленні некоректної структури рівняння система автоматично ініціює повторну генерацію варіанта.
- Алгоритми пост-обробки: фільтри валідації забезпечують педагогічно доцільне візуальне подання проміжних кроків розв'язку.

Поєднання потужних обчислювальних можливостей бібліотеки із розробленими правилами методичної фільтрації дозволяє генерувати дидактичні матеріали, що за якістю відповідають роботі досвідченого методиста, суттєво скорочуючи час на підготовку. Подальше вдосконалення системи спрямоване на оптимізацію відображення нестандартних математичних конструкцій у рідкісних випадках.

#### 6 Методика та результати впровадження системи

##### 6.1 Методичні рекомендації до застосування

Використання генератора дозволяє оптимізувати роботу вчителя, зміщуючи акцент з рутинної підготовки матеріалів на індивідуальну роботу. Основними формами застосування є:

- Контрольно-оцінювальна діяльність: швидке формування унікальних варіантів для поточного та підсумкового контролю, що мінімізує ризики академічної недоброчесності. Система дозволяє компонувати як однотипні тренувальні вправи, так і комплексні білети з завданнями різних рівнів складності.

- Методичний супровід самостійної роботи: використання згенерованих покрокових розв'язків як індивідуальних карток-інструкцій. Це дозволяє реалізувати модель «перевернутого класу», де учні самостійно опрацьовують алгоритми, а аудиторний час приділяється аналізу складних випадків.

- Корекційне навчання: генерація персоналізованих наборів завдань для роботи над помилками. Учень отримує можливість порівняти власний розв'язок із еталонним, що сприяє формуванню стійкої алгоритмічної навички та розвитку критичного мислення.

Вихідний код та зразки матеріалів доступні у репозиторії: <https://github.com/nazariyrbchynskyi/equagen>.

## 6.2 Апробація та аналіз ефективності

Педагогічний експеримент проведено на базі ліцею Чернівецької області (10 клас) на пропедевтичному етапі вивчення теми «Тригонометричні рівняння». Учні було розподілено на дві групи:

1. Контрольна група (14 учнів): самопідготовка за традиційними підручниками.

2. Експериментальна група (15 учнів): використання Telegram-бота, інтегрованого з розробленою системою, для генерації прикладів із покроковими розв'язками.

Педагогічний експеримент тривав протягом двох тижнів. На початковому етапі було встановлено однорідність груп за рівнем попередньої підготовки. Під час формувального етапу учні експериментальної групи виконували індивідуальні набори завдань, згенеровані системою, що охоплювали такі типи вправ: розв'язання базових тригонометричних рівнянь, рівнянь, що зводяться до квадратних, та однорідних рівнянь першого і другого порядків, знаходження загального розв'язку та відбір коренів на заданому проміжку.

Контрольний зріз знань проводився у формі письмової контрольної роботи, яка складалася з п'яти завдань різного рівня складності.

Результати дослідження: Аналіз контрольної роботи засвідчив, що в експериментальній групі середній бал був на 15–20% вищим, ніж у контрольній. Учні, які працювали з ботом, продемонстрували вищу алгоритмічну культуру та значно меншу кількість помилок у записах загальних формул коренів.

Одержані результати мають характер пілотного дослідження. Хоча обсяг вибірки ( $n=29$ ) обмежений рамками одного закладу освіти, виявлена позитивна динаміка успішності (на 15–20%) дозволяє зробити висновок про перспективність подальшого масштабування методики.

Опитування учасників педагогічного експерименту підтвердило, що наявність миттєвого зворотного зв'язку через покрокові пояснення підвищує автономність учнів та скорочує час на опанування типових операцій. Таким чином, апробація підтвердила ефективність системи як інструменту підвищення якості математичної освіти та успішної реалізації диференційованого підходу.

## ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШОГО ДОСЛІДЖЕННЯ

У статті розв'язано актуальну науково-прикладну задачу автоматизації створення варіативних дидактичних матеріалів з математики шляхом розробки спеціалізованого програмного забезпечення на базі мови Python та бібліотеки SymPy. Наукова новизна отриманих результатів полягає в обґрунтуванні алгоритмів «зворотного інжинірингу», які дозволили подолати проблему непередбачуваності складності коренів та гарантувати методичну відповідність згенерованих завдань чинним навчальним програмам.

Практична значущість роботи підтверджена результатами педагогічного експерименту, згідно з якими впровадження системи як засобу методичного супроводу сприяло підвищенню успішності учнів на 15–20% при одночасному скороченні часових витрат вчителя на підготовку до занять. Оцінка надійності програмного комплексу засвідчила, що реалізовані механізми валідації та адаптивного форматування ефективно нівелюють технічні anomalії машинного виводу, забезпечуючи високу якість кінцевого продукту. Подальший розвиток дослідження вбачається у розширенні бібліотеки алгоритмів на інші розділи алгебри та інтеграції розробленого ядра у веб-орієнтовані освітні платформи для масштабування отриманого досвіду.

## КОНФЛІКТ ІНТЕРЕСІВ

Автори заявляють про відсутність фінансових, особистих чи інших інтересів, що можуть розглядатися як потенційний конфлікт інтересів щодо публікації цієї статті.

## ДЖЕРЕЛА ФІНАНСУВАННЯ

Роботу виконано за відсутності фінансової підтримки з боку будь-яких організацій.

## ДОСТУПНІСТЬ ДАНИХ

Це дослідження не передбачало використання окремих наборів даних.

## ВИКОРИСТАННЯ ЗАСОБІВ ШТУЧНОГО ІНТЕЛЕКТУ

Автори використовували інструменти ШІ (зокрема Gemini) для технічного оформлення списку використаних джерел та References відповідно до вимог видання, а також для перевірки граматики й пунктуації. Автори здійснили критичну перевірку та редагування отриманого результату і несуть повну відповідальність за зміст статті.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Боднарук, С. Б., Колісник, Р. С., & Сікора, В. С. (2024). *Рівняння і нерівності в старшій школі*. Методичні рекомендації. Чернівці : Чернівецький нац. ун-т.
2. Волянська, О., & Лемішко, І. І. (2017). Використання комп'ютерних технологій під час вивчення тригонометричних рівнянь. *Актуальні проблеми теорії і методики навчання математики: до 70-річчя кафедри математики і теорії та методики навчання математики НПУ імені М. П. Драгоманова* : матеріали Міжнар. наук.-практ. конф. (м. Київ, 11–13 трав. 2017 р.). Київ : Вид-во НПУ імені М. П. Драгоманова, 100–101.
3. Істер, О. С., & Єрміна, О. В. (2018). *Алгебра і початки аналізу : (профіль. рівень)* : підруч. для 10-го кл. закл. заг. серед. освіти. Київ : Генеза.
4. Мерзляк, А. Г., Номіровський, Д. А., Полонський, В. Б., & Якір, М. С. (2018). *Алгебра і початки аналізу : початок вивчення на поглибленому рівні з 8 кл. : профільний рівень* : підруч. для 10 кл. закл. заг. серед. освіти. Харків : Гімназія.
5. Міністерство освіти і науки України (2017а). *Навчальна програма з математики (алгебра і початки аналізу та геометрія) для учнів 10–11 класів загальноосвітніх навчальних закладів*. Рівень стандарту. URL: <https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-10-11-klas/2018-2019/matematika-riven-standartu.docx>
6. Міністерство освіти і науки України (2017б). *Навчальна програма з математики для учнів 10–11 класів (початок вивчення на поглибленому рівні з 8 класу) загальноосвітніх навчальних закладів*. URL: <https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-10-11-klas/2018-2019/matematika-poglibl-rivenfinal.docx>
7. Міністерство освіти і науки України (2017с). *Навчальна програма з математики для учнів 10–11 класів загальноосвітніх навчальних закладів*. Профільний рівень. URL: <https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-10-11-klas/2018-2019/matematika-profilnij-rivenfinal.docx>
8. Нелін, Є. П. (2018). *Алгебра і початки аналізу (профільний рівень)* : підруч. для 10 кл. закл. загал. серед. освіти. Харків : Ранок.
9. Ali, A., & Khusro, S. (2024). SA-MEAS: Sympy-based automated mathematical equations analysis and solver. *SoftwareX*, 25, 101596. <https://doi.org/10.1016/j.softx.2023.101596>
10. Chan, K. W., Ali, F., Park, J., Sham, K. S. B., Tan, E. Y. T., Chong, F. W. C., Qian, K., & Sze, G. K. (2025). Automatic item generation in various STEM subjects using large language model prompting. *Computers and Education: Artificial Intelligence*, 8, 100344. <https://doi.org/10.1016/j.caeai.2024.100344>
11. Fitri, P., Hartono, Y., & Meryansumayeka, M. (2025). Learning proof of trigonometric identities with ChatGPT. *Journal of Honai Math*, 8(1), 43–56. <https://doi.org/10.30862/jhm.v8i1.755>
12. Karbasi, K., Hong, K., Samadi, M. A., & Pottie, G. (2025). *Multi-Agent Collaborative Framework For Math Problem Generation*. 613–618. <https://doi.org/10.5281/zenodo.15870246>

## REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Bodnaruk, S. B., Kolisnyk, R. S., & Sikora, V. S. (Eds.). (2024). *Rivniannia i nerivnosti v starshii shkoli. Metodichni rekomendatsii* [Equations and inequalities in high school: Methodological recommendations]. Chernivtsi National University. (in Ukrainian)
2. Volianska, O., & Lemishko, I. I. (2017). Vykorystannia kompiuternykh tekhnolohii pid chas vuvchennia trygonometrychnykh rivnian [The use of computer technologies in studying trigonometric equations]. In *Aktualni problemy teorii i metodyky navchannia matematyky* (pp. 100–101). National Pedagogical Dragomanov University. (in Ukrainian)
3. Ister, O. S., & Yerhina, O. V. (2018). *Alhebra i pochatky analizu (profilnyi riven): Pidruchnyk dlia 10 klasu zakladiv zahalnoi serednoi osvity* [Algebra and beginnings of analysis (profile level): Textbook for grade 10]. Geneva. (in Ukrainian)
4. Merzlyak, A. H., Nomirovskiy, D. A., Polonskiy, V. B., & Yakir, M. S. (2018). *Alhebra i pochatky analizu: Profilnyi riven (pochatok vuvchennia z 8 klasu): Pidruchnyk dlia 10 klasu zakladiv zahalnoi serednoi osvity* [Algebra and beginnings of analysis: Profile level]. Himmaziia. (in Ukrainian)
5. Ministry of Education and Science of Ukraine. (2017a). *Navchalna prohrama z matematyky (alhebra i pochatky analizu ta heometriia) dlia uchniv 10–11 klasiv zakladiv zahalnoi serednoi osvity. Riven standartu* [Mathematics curriculum for grades 10–11. Standard level]. URL: <https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-10-11-klas/2018-2019/matematika-riven-standartu.docx> (in Ukrainian)
6. Ministry of Education and Science of Ukraine. (2017b). *Navchalna prohrama z matematyky dlia uchniv 10–11 klasiv (pochatok vuvchennia na pohlyblyenomiu rivni z 8 klasu)* [Mathematics curriculum for grades 10–11 (advanced level)]. URL: <https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-10-11-klas/2018-2019/matematika-poglibl-rivenfinal.docx> (in Ukrainian)
7. Ministry of Education and Science of Ukraine. (2017c). *Navchalna prohrama z matematyky dlia uchniv 10–11 klasiv. Profilnyi riven* [Mathematics curriculum for grades 10–11. Profile level]. URL: <https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-10-11-klas/2018-2019/matematika-profilnij-rivenfinal.docx> (in Ukrainian)
8. Nelin, Ye. P. (2018). *Alhebra i pochatky analizu (profilnyi riven): Pidruchnyk dlia 10 klasu zakladiv zahalnoi serednoi osvity* [Algebra and beginnings of analysis (profile level)]. Ranok. (in Ukrainian)
9. Ali, A., & Khusro, S. (2024). SA-MEAS: Sympy-based automated mathematical equations analysis and solver. *SoftwareX*, 25, 101596. <https://doi.org/10.1016/j.softx.2023.101596>
10. Chan, K. W., Ali, F., Park, J., Sham, K. S. B., Tan, E. Y. T., Chong, F. W. C., Qian, K., & Sze, G. K. (2025). Automatic item generation in various STEM subjects using large language model prompting. *Computers and Education: Artificial Intelligence*, 8, 100344. <https://doi.org/10.1016/j.caeai.2024.100344>
11. Fitri, P., Hartono, Y., & Meryansumayeka, M. (2025). Learning proof of trigonometric identities with ChatGPT. *Journal of Honai Math*, 8(1), 43–56. <https://doi.org/10.30862/jhm.v8i1.755>
12. Karbasi, K., Hong, K., Samadi, M. A., & Pottie, G. (2025). *Multi-Agent Collaborative Framework For Math Problem Generation*. 613–618. <https://doi.org/10.5281/zenodo.15870246>

| Матеріал надійшов до редакції: 28.01.2026 р. | Прийнято до друку: 05.03.2026 р. | Опубліковано: 30.04.2026 р. |

