doi: 10.32620/oikit.2024.102.07

УДК 004.58

А. Г. Чухрай, Т. Л. Столяренко, О. О. Євдокимов, В. А. Дем'яненко

# Можливості використання інтелектуальних навчальних систем (ITS) в ЗВО для курсів вищої математики

*Національний аерокосмічний університет ім. М. Є. Жуковського «Харківський авіаційний інститут»*

Розглянуто сучасний стан рівня знань з математики серед випускників шкіл як в Україні, так і у світі. Виявлено негативну тенденцію, що вимагає термінових змін у підходах до навчання. У цій роботі досліджено можливості використання інтелектуальних комп'ютерних навчальних програм для зменшення негативного впливу на якість математичної освіти. Проведено аналіз існуючих моделей таких програм і розглянуто їх застосування у викладанні математики. Як методи дослідження використано математичне моделювання та аналіз відкритих джерел. Результати роботи включають розробку математичних моделей інтелектуального комп'ютерного навчання математики. Запропоновано класифікацію задач для навчання на дві підгрупи: прості задачі (на застосування алгоритмів) і складні задачі (на складання алгоритмів). Розглянуто ключові відмінності у підходах до реалізації моделей навчання для кожного з класів задач. Особливу увагу приділено задачам на складання алгоритмів, зокрема задачам на доведення теорем і виведення формул. Запропоновано новий підхід до навчання розв'язання таких задач, який враховує специфіку їхньої структури та складності. Також досліджено можливості використання генеративного штучного інтелекту для автоматизації створення унікальних задач і баз даних. Запропоновано комбінувати алгоритмічну точність і лексичну гнучкість моделей генеративного штучного інтелекту для побудови повної моделі будь-якої навчальної задачі. Проведено аналіз існуючих проектів і технологій у цій сфері, розроблено порівняльну характеристику їхніх переваг і недоліків. Виявлено перспективний і слабо досліджений напрям — створення математичного користувацького інтерфейсу для систем автоматичної перевірки доведення теорем, що є актуальним для покращення якості математичної освіти. Результати дослідження відкривають нові можливості для вдосконалення інтелектуальних навчальних систем, а також мають практичну цінність для інженерів і підприємців. Робота пропонує науковцям базу для подальших досліджень у сфері автоматизації математичної освіти та її адаптації до сучасних викликів.

***Ключові слова***: КН; ІКНП; PISA; ITS; ШІ; ITP; LLM; система перевірки доведення; метаматематика.

## Вступ

Сучасний світ характеризується величезними обсягами інформації, що зростають експоненціально, швидким технологічним розвитком та новими викликами перед людством. В основі багатьох системних рішень, наукових відкриттів та технологічних проривів лежить дисципліна математики. Тому підтримка високого рівня компетенції із цієї дисципліни серед школярів, студентів, професіоналів та в цілому населення напряму впливає на успіх та пристосованість окремих індивідів та всього суспільства до викликів сучасності.

Однак, як буде наведено далі в дослідженні, актуальна національна та світова тенденції рівня компетенції з математики має зворотній характер, що створює додаткові виклики і проблеми, які намагаються вирішити автори даної статті. В роботі чітко викладаються основні причини даних тенденцій та фактори, що мають вплив. Зокрема часта зміна навчальних програм, недостатня підготовка викладачів та психологічні впливи, як у викладачів, так і в студентів, наприклад математична тривожність перед *складними задачами* (math anxiety). В умовах пандемії COVID-19 ситуація значно погіршилась, оскільки студенти втратили можливість повноцінного очного навчання. Як показують дані постковідного періоду, ситуація не змінилася на краще.

Враховуючи дані виклики і тенденції автори статті пропонують шляхи подолання цих проблем і стверджують, що впровадження сучасних засобів адаптивної інтелектуальної комп’ютерної підтримки навчання здатне змінити негативну тенденцію та усунути або нейтралізувати негативні фактори. Це твердження підкріплюється доказами ефективності застосування інтелектуальних комп’ютерних навчальних програм в навчальних закладах.

В наступних розділах роботи розглядаються різновиди сучасних моделей інтелектуальної комп’ютерної підтримки навчання математики, пропонуються покращення цих моделей за рахунок використання великих мовних моделей (LLM) як от ChatGPT, та наводиться огляд існуючих розробок.

Останній розділ роботи присвячений огляду інтелектуальної навчальної комп’ютерної програми, що розробляється в Національному аерокосмічному університеті ім. М. Є. Жуковського «ХАІ», та актуальним задачам і відкритим питанням, які стануть корисними науково-педагогічній спільноті, інженерам та керівникам продуктів сфери освітніх інформаційних технологій (EdTech).

## 1. Тенденції та проблематика

Підтримка високого рівня математичних компетенцій серед школярів, студентів та професіоналів, як окремих індивідуумів, напряму впливає на їх успіх і адаптацію до викликів сучасного світу, а також на успіх суспільства і цивілізації загалом [1]. Можна скористатися дослідженнями, що підтверджують важливість знання математики та сучасних технологій для освітнього, професійного та соціального успіху [13].

За даними досліджень, рівень математичної компетентності студентів тісно пов'язаний із їхніми майбутніми досягненнями у наукових і технологічних сферах. Наприклад, дослідження, що аналізують результати PISA (Programme for International Student Assessment), показують, що студенти, які досягають успіху в математиці, мають вищі шанси на успіх у науково-технічних дисциплінах та адаптацію до технологічних інновацій у майбутньому [11, 12]. Інші дослідження демонструють, що студенти з високим рівнем математичної підготовки легше адаптуються до технологій та інновацій, що є критично важливими в сучасному цифровому світі [7, 14].

Дослідження також підтверджують, що країни з високим рівнем математичної підготовки серед населення здатні швидше впроваджувати технологічні інновації та вирішувати складні виклики. Це свідчить про те, що підтримка високого рівня математичної освіти є ключовою умовою для успіху в глобальному масштабі. Таким чином, підтримка високого рівня математичної підготовки є необхідною для успішного вирішення сучасних викликів і сприяє технологічному та науковому прогресу як на індивідуальному, так і на суспільному рівнях.

Спираючись на наявні відкриті дані та дослідження [2, 3, 8] можна впевнено стверджувати, що якість підготовки школярів в Україні та світі почала зменшуватись. Розподіл результатів учасників НМТ за виконання субтесту з математики (у шкалі 100–200) у 2024 році наведено на рис. 1.

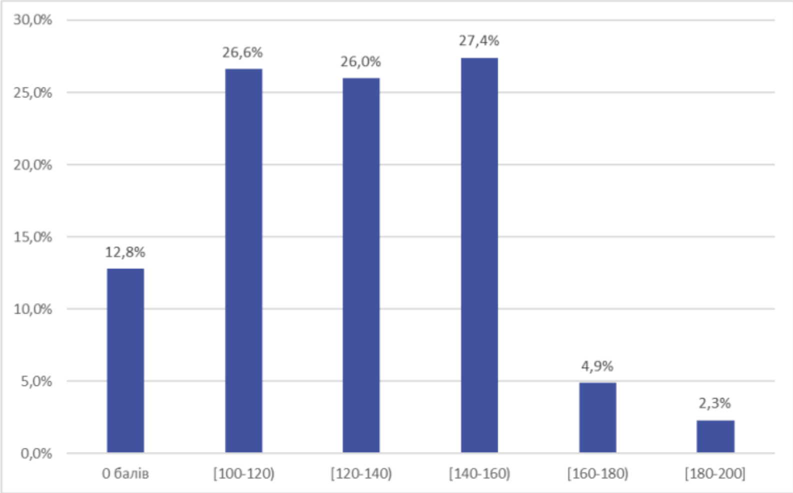



Рис. 1. Розподіл результатів учасників НМТ з математики за 2024 рік

Частка тих, хто не набрав мінімальний бал або 100-120 балів значно збільшилася відносно попереднього року, дані якого можна побачити на рис. 2. А частка “відмінників”, які набрали 180-200 балів зменшилась на третину.

Ці значення показують значну негативну тенденцію відносно розподілів результатів НМТ за виконання субтесту з математики 2023 року та середніх результатів за цей рік [8], аналіз якого був проведений в дослідженні [2]. Навіть негативні статистичні показники результатів в 2023 році були кращими, ніж результати в цьому році.

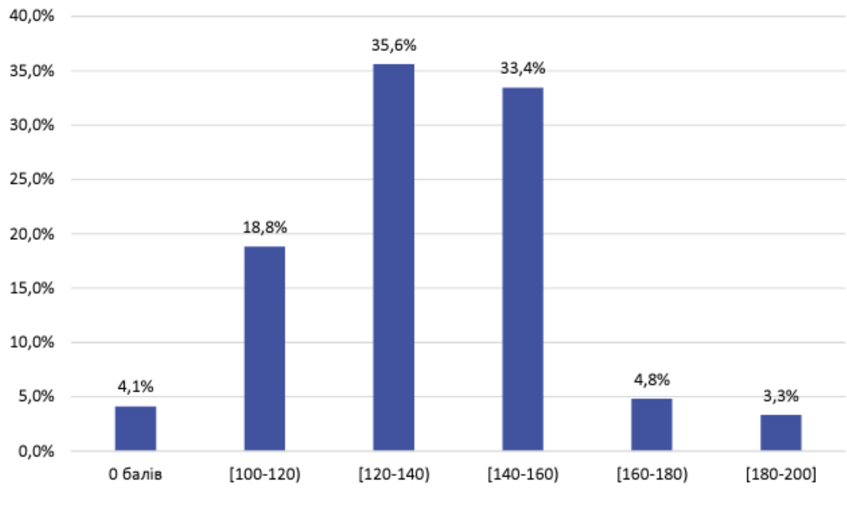



Рис. 2. Розподіл результатів учасників НМТ з математики за 2023 рік

Вище були наведені дані лише по ситуації в Україні, далі розглянемо ситуацію в Сполучених Штатах Америки.

Національний центр статистики освіти США кожні 4 роки проводить оцінку успішності 9- ,13- та 17-річних студентів та готує звіт National Assessment of Educational Progress (NAEP), відомий як "Національний звіт про успішність".

Згідно з результатами NAEP за 2022-2023 роки, середні бали з математики серед учнів 4-го та 8-го класів знизилися до найнижчого рівня за попередні два десятиліття. Середній бал з математики для 13-річних учнів у 2023 році був на 5 балів вищим, ніж у 1973 році, у перший рік оцінювання з математики довгостроковий тренд - Long-Term Trend (LTT). Проте порівняно з попереднім оцінюванням LTT у 2020 році середній бал у 2023 році був на 9 балів нижчим. Середні бали наводяться за математичною шкалою NAEP LTT, яка коливається від 0 до 500 [3].

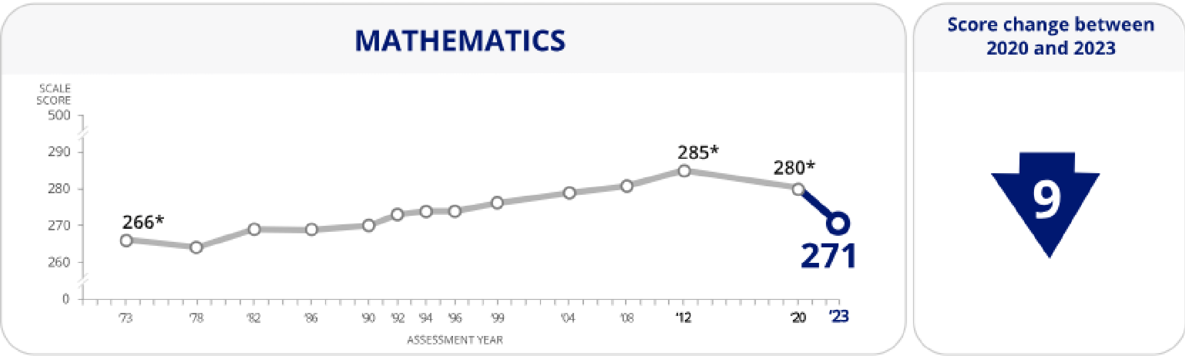



Рис. 3. Тренд середніх балів з математики NAEP для 13-річних учнів

Розглянемо фактори, що могли призвести до цієї тенденції. Серед основних можна виділити:

- **пандемію COVID-19**, що призвела до дистанційної форми навчання; раптовий перехід до онлайн-освіти призвів до дезорганізації навчального процесу; багато учнів не були готові до самостійного навчання в домашніх умовах, що знизило ефективність засвоєння матеріалу; відсутність безпосереднього контакту з вчителями та однокласниками ускладнила можливість отримувати швидкий зворотний зв'язок та підтримку.
- **психологічні фактори**, наприклад в дослідженні [10] автори показують, що люди з математичною тривожністю схильні уникати математичних завдань, що негативно впливає на їхні результати, одна із причин – це відсутність позитивного підкріплення протягом навчання, а також стрес та постійна психологічна напруга, що могли бути спричинені через пандемію та соціальні обмеження.
- **проблеми освітньої системи**, такі як недостатня підготовка вчителів, застарілі методики викладання, що можуть бути неефективними для сучасних учнів, особливо в дистанційному форматі, та великий обсяг матеріалу без достатньої практичної спрямованості, що може знижувати інтерес учнів до математики.
- **культурні та мотиваційні аспекти**, наприклад стереотипи щодо математики, що математика є складною і не кожному під силу, що може відштовхувати учнів від вивчення предмету; та відсутність практичної застосовності, якщо учні не бачать практичного застосування математичних знань у реальному житті, їх мотивація до вивчення знижується.

Ці фактори повинні бути враховані при розробці рішень для зупинки негативних тенденцій в рівні знань математики серед студентів. Але якщо узагальнити вищезгадані фактори, основний виклик – це можливість підтримки високого рівня знань математики при дистанційному навчанні, мінімізуючи або взагалі усуваючи негативні впливи та навіть використання цих впливів на користь розвитку та покращення студента в математиці.

Наша команда пропонує рішення у вигляді створення інтелектуальної комп’ютерної навчальної програми (ІКНП), яка враховує сучасні виклики та нівелює спад рівня знань математики серед школярів, та впровадження її в університетах та коледжах України.

Чому фокус робиться саме на вищій математиці? Оскільки спроби створення рішень лише для шкіл потребують значних системних змін в освіті та їх впровадження по всіх школах, це несе за собою більші витрати на зміни. Також вони мають бути ініційовані урядовими підрозділами з питань освіти. Впроваджувати зміни в університеті легше, економічно обґрунтовано та продуктивно з точки зору величини впливу на всю освітню систему. Це має місце бути, оскільки університети мають потужну науково-технічну базу досвідчених викладачів, просунутих професіоналів, що мають фундаментальні знання дисципліни. Щодо наслідків та впливу цих змін, вони є не лише прямими, такими як покращення навчальної програми та адаптація слабких за рівнем знань студентів(колишніх школярів) до університетської програми вищої математики та фундаментальне повторення і засвоєння невивчених концепцій шкільної математики, але і непрямими, такими як покращення рівня знань математики серед населення в цілому, та випуск з університету більш якісно підготовлених майбутніх вчителів шкільної математики, що в свою чергу має прямий вплив на покращення рівня знань математики серед школярів.

## 2. Моделювання та шляхи вирішення

Досягнення широкої доступності розуміння фундаментальних концепцій дисципліни є основною метою створення інтелектуальної комп’ютерної навчальної програми, оскільки основні виклики, що стоять перед студентами, які вступають до вишів – є концептуальне нерозуміння математичної термінології, теорії та її застосування на практиці. В нашому дослідженні ми робимо фокус на розвитку вмінь вирішувати задачі, оскільки основні вимоги, за якими оцінюються рівень знань студента – це вміння вирішувати математичні задачі. Враховуючи це ми розділяємо задачі на прості та складні. Робиться таке розділення з метою спрощення створення інтелектуальної комп’ютерної навчальної програми та контролем обмежень, які вона може мати. Варто показати різницю між простими математичними задачами та складними, але перед цим, наведемо деякі визначення.

Розглянемо основні напрями розробки ІКНП. Як показано в роботі [15] американського дослідника Курта ВанЛена більшість ІКНП можуть складатись із двох циклів – зовнішнього і внутрішнього. Зовнішній цикл відповідає за вибір наступної задачі для студента, внутрішній – призначений для покрокової допомоги студенту у вирішенні конкретної задачі. В цьому дослідженні ми більше зробили акцент на можливостях внутрішнього циклу, оскільки саме він найбільше відповідає за можливість студентів розуміти концепції математики. В той час як зовнішній цикл відповідає за послідовність задач та за рівень складності матеріалу.

Одним із ключових термінів є «алгоритм». Цей термін співпідає із загальновідмими визначеннями, наприклад, наведеними в фундаментальній праці Л. Н. Ланди [4]. Під алгоритмом розуміють точний загальнодоступний опис послідовності елементарних операцій для вирішення завдання, що належить до певного класу. Крім того, алгоритми характеризуються рядом основних властивостей: детермінованість, масовість, результативність [5].

*Детермінованість* означає, що команди, що входять в алгоритм, виключають випадковість та при всіх однакових вхідних даних призводять до однакового результату незалежно від того, при яких умовах він виконується.

*Масовість* виражається в тому, що алгоритм призначений для рішення не лише одного завдання з деякими вхідними значеннями, а для рішення цілого класу завдань з різними вхідними даними.

*Результативність* розуміється як цілеспрямованість алгоритма на отримання деякого потрібного результату.

Повертаючись до порівняння математичних задач за складністю зазначимо.

***Прості задачі*** — це завдання, які можна розв'язати з використанням базових математичних знань або елементарних алгоритмів. Для їх розв'язання часто не потрібно багато часу або складних концепцій. Їх можна розв'язати, використовуючи базові арифметичні операції (додавання, віднімання, множення, ділення). Вони зазвичай не вимагають складних логічних міркувань або маніпуляцій з формулами. Прості задачі часто зводяться до алгоритмів, які мають лінійну або низьку обчислювальну складність. Зазвичай вони використовуються для навчання основним методам і правилам та для перевірки їх знань.

*Прості задачі* основані на правилах, які можна зручно зберегти в базі даних та на простих алгоритмах. Наведемо визначення:

$$R = \left\{ r_i \right\}_{i=1}^{n}, \tag{1}$$

– набір правил та концепцій, знання яких перевіряється в задачі та які можна зберегти в базі даних.

$$Input = \left\{ inp_i \right\}_{i=1}^{m}, \ inp_i \in \mathbb{R}^n \tag{2}$$

– множина вхідних даних задачі, елементи якої як правило є числами, або наборами чисел.

$$Answer = \left\{ ans_i \right\}_{i=1}^{k}, \tag{3}$$

– множина відповідей для задачі.

$$Algorithm = \left\{ A_i \right\}_{i=1}^{s}, \tag{4}$$

– множина рішень задачі. Як правило для простих задач не існує багато варіантів розв’язку, тому $S << 10$.

Запишемо математичну модель задачі:

$$Task = \left( R, Given, Answer, Input, Algorithm \right), \tag{5}$$

$$Given = \left\{ g_i \right\}_{i=1}^{l}, \tag{6}$$

– множина можливих постановок однієї тієї самої задачі.

Модель постановки задачі на знання правил (теорія) в загальному вигляді можна записати наступним чином:

$$\begin{gathered} DefinitionTask = \left( R, Given, Answer \right) \\ \exists r_i \in R, \ r_i \in Answer \end{gathered}, \tag{7}$$

– також ця модель є представленням перевірки знань термінів чи визначень.

Запишемо модель постановки задачі на знання застосування правила для конкретної життєвої ситуації чи задачі:

$$\begin{gathered} PracticeDefinitionTask = \left(R, Given, Answer, Algorithm\right) \\ \forall g \in Given\ \exists r_i \in R:\ r_i \in Answer,\ \exists A \ \in Algorithm \end{gathered}, \quad (8)$$

Модель рішення будь-якої простої задачі в загальному вигляді наведена в формулі (9):

$$\begin{gathered} A: Input \to Answer, \\ \forall inp_i \in Input\ \exists ans_j \in Answer: ans_j = A\left(inp_i\right) \end{gathered}, \quad (9)$$

Перелічені моделі дозволяють наочно побачити, що головне, чому повинен навчитися студент на простих задачах, це саме знання правил $R$ та розуміння і вміння застосовувати алгоритми $A$ базуючись на конкретних правилах.

*Приклади*:

- рівняння з однією змінною: $23 + x = 47$.
- задача на визначення площі квадрата з довжиною сторони $a = 5$: $S = a^2 = 5^2 = 25$.
- і так далі.

***Складні задачі*** – це завдання, для розв'язання яких потрібне розуміння більш глибоких або абстрактних концепцій. Вони можуть потребувати розв'язання кількох етапів або використання просунутих математичних методів. Складні задачі зазвичай вимагають не лише арифметичних операцій, а й вміння використовувати теореми, логіку або певні математичні докази. Їх розв'язання часто потребує кілька кроків, де на кожному етапі потрібно застосовувати інші методи або розрахунки. Для їх розв'язання може знадобитися знання різних математичних концепцій, наприклад диференціальних рівнянь, теорії ймовірностей, математичного аналізу тощо. Їх вирішення може включати алгоритми з експоненціальною або поліноміальною складністю, які важко або неможливо розв'язати з використанням базових методів.

Математична модель складної задачі записується наступним чином:

$$\begin{gathered} ComplexTask = \left(R, Given, Answer, Input, Algorithm\right) \\ \forall g \in Given\ \exists theoryAns \in Algorithm \bigcup R: \\ \forall inp \in Input\ \exists ans \in \square\ ,\ Answer = Algorithm \bigcup R \bigcup \square \end{gathered}, \quad (10)$$

*Приклади*:

- задача комівояжера (NP-повна задача): знайдіть найкоротший шлях, який проходить через кілька міст, відвідуючи кожне рівно один раз і повертаючись до початкової точки.
- теореми з доведенням: доведіть, що сума кутів у трикутнику дорівнює 180 градусам.
- задача на максимізацію прибутку з обмеженнями (лінійне програмування): оптимізуйте виробництво на заводі для отримання максимального прибутку, маючи задані обмеження.
- і так далі.

Варто зазначити, що термін *алгоритм* стає основною складовою для обох типів задач, які є вичерпним набором для навчання математики та тестування рівня знань, виходячи із обмежень сучасної обчислювальної техніки.

Узагальнивши порівняння простих та складних задач, можна зробити висновок, що прості задачі потребують перевірки відповіді на ідентичність числу або твердженню, правилу. В той час як складні задачі потребують складної перевірки відповіді, наприклад тестування алгоритму або формальну верифікацію доведення теореми або алгоритму.

Автоматизація навчання рішення складних задач є відповідно складною задачею для будь-якого інженера. Тому при реалізації таких моделей оптимально використовувати існуючі реалізації моделей формальної верифікації.

Розглянемо як навчання вирішенню цих двох типів задач може бути реалізоване за допомогою інтелектуальних комп'ютерних навчальних програм.

## 3. Реалізація моделей

Оскільки розглянуті моделі покривають більшість задач із навчання математики, розглянемо їх реалізацію та застосування. Виходячи з моделей (7), (8), (9), запропонуємо класифікацію параметрів моделей.

Множину правил та концепцій $R$ можна розглядати як набір елементів, записаних у вигляді тексту, картинок, формул, теорем, визначень, які розкривають та пояснюють повну модель конкретної задачі.

Множина постановок задачі $Given$ складається із елементів, які можуть бути записані у вигляді тексту, картинок, формул, теорем, визначень, але містять питання та проблему, яку необхідно вирішити, або значення, яке необхідно знайти. Також в цій множині відсутня інформація, яка може вказувати на повний набір правил та концепцій задачі $R$.

Множина відповідей $Answer$ складається із елементів, які можуть бути записані у вигляді тексту або картинок або формул або теорем, визначень, але цю множину можна поділити на дві підмножини правильних відповідей та неправильних, які будемо перевіряти за допомогою функції (11):

$$\begin{gathered} IsCorrect : Answer \to Boolean \\ \forall ans \in Answer\ \exists result \in Boolean :, \\ IsCorrect(ans) = result \end{gathered} \tag{11}$$

елементи множини $Answer$ можуть бути також і числами у випадку практичних задач.

Множина вхідних даних задачі $Input$ як правило є числами, в рідких випадках формулами.

Множина рішень задачі $Algorithm$ є множиною функцій, що на вході приймають елемент множини $Input$, а на виході отримують елемент множини $Answer$. Ці функції можуть бути реалізовані як за допомогою сучасних мов програмування, так і за допомогою систем перевірки формального доказу.

Пропонується що модель задачі повинна наповнюватись асистентами-викладачами згідно їх робочої програми за курсом, але система повинна бути створена так, щоб мінімізувати час, який необхідно витратити на це та час на використання. В сучасному академічному середовищі ефективне використання робочого часу викладачів вищої школи є ключовим фактором для забезпечення високої якості освіти та наукового прогресу. Звільнення часу від рутинних завдань, таких як створення варіантів задач та перевірка робіт, дозволяє викладачам більше зосередитися на наукових дослідженнях у своїй галузі, оптимізації навчальних курсів та розробці складних, нетривіальних задач.

По-перше, активна наукова діяльність викладачів сприяє актуалізації навчального матеріалу та інтеграції новітніх досліджень у навчальний процес. Це не лише підвищує рівень засвоєння знань студентами, але й стимулює їхню зацікавленість у предметі та мотивацію до дослідницької роботи. Включення в курси сучасних наукових досягнень розширює горизонти студентів.

По-друге, оптимізація курсів на основі аналізу даних про успішність та зворотного зв'язку від студентів дозволяє викладачам підвищити ефективність навчання.

По-третє, автоматизація рутинних процесів, таких як перевірка завдань або створення їхніх варіантів, за допомогою інтелектуальних комп'ютерних програм дозволяє викладачам ефективніше використовувати свій час. Це не лише підвищує продуктивність праці, але й покращує якість освітнього процесу, оскільки викладачі можуть більше уваги приділяти індивідуальній роботі зі студентами та науковому наставництву.

В назві статті є термін «інтелектуальні навчальні програми». Покажемо де саме ми плануємо використати інтелектуальний підхід. Враховуючи специфіку зовнішнього та внутрішнього циклу, в цій роботі не станемо розглядати підходи до зовнішнього циклу, оскільки це окрема складна тема, що потребує окремого дослідження.

Візьмемо параметри моделі задачі, а саме параметр $R$ – множину правил та концепцій. Саме ця множина і відповідає за зовнішній цикл, оскільки в ній враховані всі концепції, теми та визначення, які учень повинен освоїти та знати, щоб вирішити певну задачу. Тому учня не можна одразу переводити на задачу з великою множиною тем та концепцій, не створюючи плавний перехід в навчанні від простих концепцій та їх кількості до складних. Також необхідно відстежувати рівень кожного учня та відповідно підбирати складність задачі. Лише наведемо приклад на рисунку 4 підмножину супермножини $R$, з елементів якої формується множина $R$ для задачі [2].

Перейдемо до множини постановок задач $Given$. Її необхідно вручну створювати асистентам-викладачам, вводячи в текстове поле різні постановки однієї і тієї самої задачі. Але тут є можливість автоматизувати цей процес за допомогою генеративних моделей штучного інтелекту. Подібні варіанти використання наводяться в дослідженні [16]. Наведемо приклади таких постановок задачі:

**Задача на обчислення інтеграла:**

- Обчисліть визначений інтеграл функції $f(x)=x^2$ на інтервалі $[1;4]$.
- Знайдіть площу під кривою $y=x^2$ між точками $x=1$ та $x=4$.
- Визначте значення інтеграла $\int_1^4 x^2 dx$.

- Розрахуйте суму нескінченно малих елементів площі під графіком функції $y = x^2$ від $x = 1$ до $x = 4$.
- І т.д.

Дані варіанти постановки задачі були згенеровані великою мовною моделлю ChatGPT o1-preview. Але іноді генеративні моделі можуть давати помилки, тому дуже важливо, щоб викладач-асистент мав можливість перевірити та затвердити конкретну постановку.

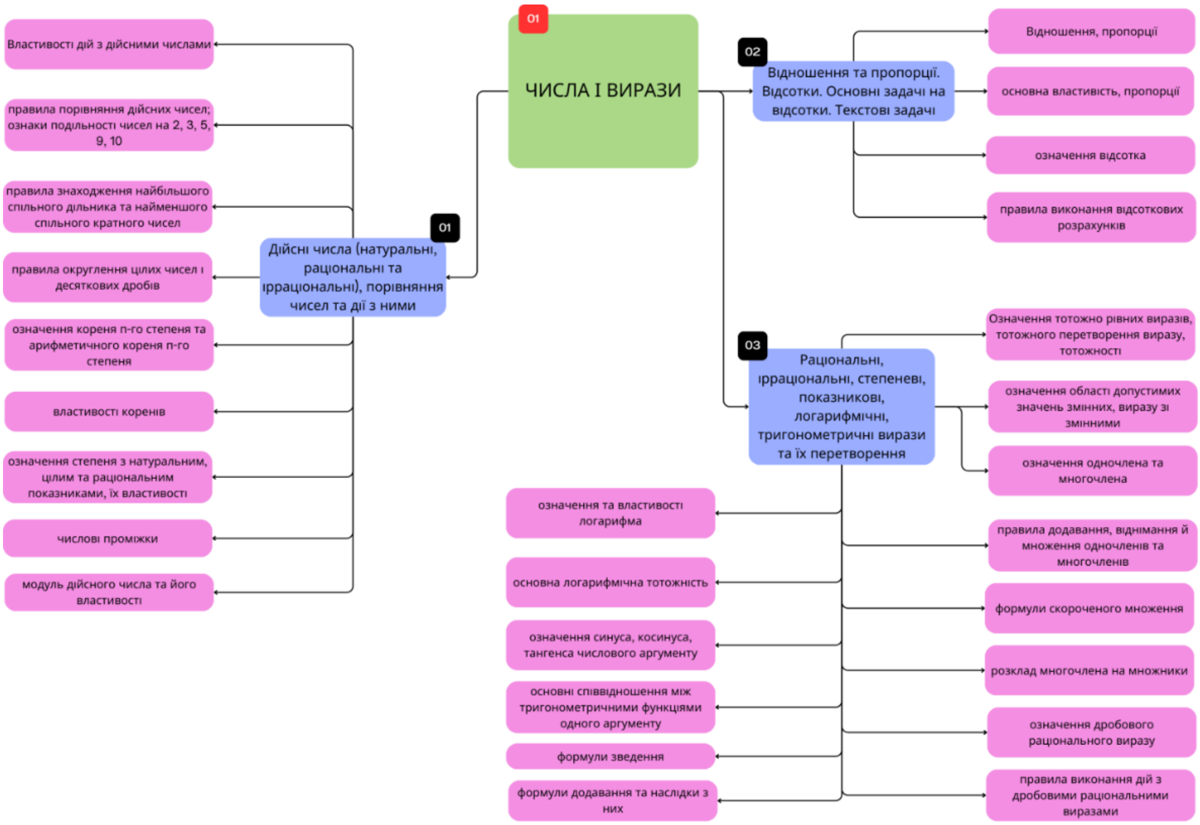



Рис. 4. Підмножина всіх математичних правил та концепцій, що враховані в системі [2]

Оскільки великі мовні моделі (LLM) не завжди здатні створювати моделі задач у загальному вигляді, особливо для складних задач, але добре справляються в генерації текстових постановок, то обмежимось використанням великих мовних моделей лише для генерації варіацій текстової складової задачі. Для $DefinitionTask$ та $PracticeDefinitionTask$ (задач на визначення), можна обмежитись лише розглянутим підходом. У випадку $PracticeDefinitionTask$ генерація $Algorithm$ є умовною, оскільки прямо не впливає на правильну відповідь.

Інакша ситуація виникає з практичними задачами, необхідною умовою якої є надійно працююча модель для будь-якого елементу множини $Input$. Правильна відповідь на попередню задачу-приклад є $21$. Однак це значення може бути іншим під впливом різних елементів $Input$. Введемо поняття алгоритму, що генерує вхідні дані та відповідь. Цей алгоритм є основою при створенні нових вхідних даних для задачі. Саме його разом з алгоритмом самої задачі $Algorithm$ і пропонується вводити викладачу в систему за допомогою задання обмежень та/або блок-схем/псевдокоду або мови програмування. За допомогою генеруючого алгоритму формується множина $Input$ , після чого використовуючи $Algorithm$ – множина $Answer$ . На рисунку 5 зображена блок-схема використання генеруючого алгоритму [2].

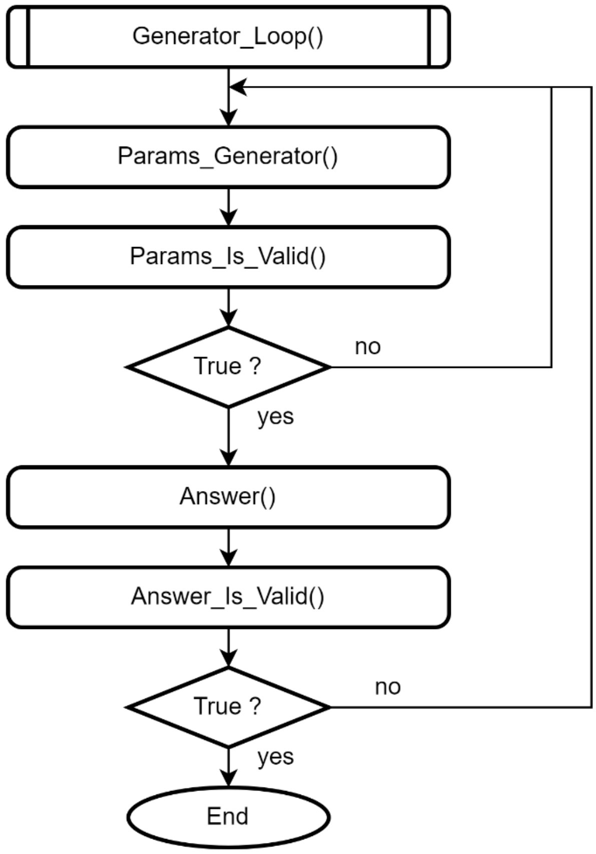



Рис. 5. Блок-схема використання генеруючого алгоритму

Generator_Loop() визиває метод Params_Generator(), який використовує генератори псевдовипадкових чисел для генерації значень параметрів. При цьому користувач-учитель може задати тип значення (ціле або дійсне число) та діапазони припустимих значень. Якщо всі параметри є незалежними, їх можна описувати у будь-якому порядку. Якщо ж присутні залежні параметри, спочатку оцінюються незалежні, а потім описуються вирази для розрахунку залежних параметрів [2].

Але деякі поєднання параметрів можуть виявитись неприпустимими або призводити до некоректних результатів. Для цього передбачені дві перевірочні функції. Функція Params_is_Valid() перевіряє параметри на відповідноть певним умовам одразу після генерації. Після викликаючи функцію Answer() запускається $Algorithm$ . А потім викликаємо функцію Answer_is_Valid() для перевірки коректності отриманої відповіді. Якщо будь-яка з цих функцій повертає False або завершується помилкою, цикл повторюється [2].

При використанні цієї моделі, учень вводить значення відповіді на завдання, яке перевіряється на правильність та відповідність $IsCorrect(ans) = True$.

Розглянутий підхід може бути застосованим для простих задач і бути реалізованим на будь-якій загальновживаній мові програмування створення клієнт-серверних додатків.

Тепер розглянемо випадок для складних задач, де необхідно знайти або скласти $Algorithm$. Можна сказати впевнено, що ці завдання повинні складатись з особливою увагою асистентами-викладачами. А саме постановка задачі та вхідні значення. Основним викликом є перевірка правильності відповіді на складну задачу.

Одним із підходів є підхід написання тестів аналогічних тестам для промислового програмного забезпечення, цей підхід добре описаний в роботі [6]. Він добре підійде для складних задач з обмеженою кількістю та варіацією вхідних та вихідних значень алгоритму. Однак і в такому випадку існує ймовірність обходу системи перевірки учнем та надання неправильної відповіді, яку система розцінить як правильну.

Іншим, найбільш перспективним підходом, хоч і складним в реалізації, є формальна перевірка алгоритму за допомогою систем та інструментів перевірки доказу (*proof checking*). Історія даних інструментів стосується розділу математичної логіки – *метаматематики* [19]. В 1931 році австрійський математик Курт Гедель показав, що в будь-якій досить сильній аксіоматичній системі є істинні твердження, які не можуть бути доведені в цій же системі [17]. Ця тема була далі розроблена в 1930-х роках Алонзо Черчом [18] і Аланом Тюрінгом , які, з одного боку, дали два незалежних, але еквівалентних визначення обчислюваності , а з іншого – навели конкретні приклади нерозв'язних питань, для яких неможливо побудувати алгоритм рішення. Дані задачі не розглядаються в системі навчання. Залежно від логіки, що лежить в основі, проблема визначення істинності формули, теореми, алгоритму варіюється від тривіальної до неможливої. У загальному випадку пропозиційної логіки проблема є розв'язною, але NP-повною. Таким чином, вважається, що для загальних задач доведення існують лише алгоритми з експоненційним часом виконання. Однак для обчислення предикатів першого порядку теорема про повноту Геделя стверджує, що теореми (твердження, які можна довести) є точно семантично істинними добре сформованими формулами, тому істинні формули є обчислюваними: теоретично за наявності необмежених ресурсів будь-яку істинну формулу можна зрештою довести. Однак хибні формули (ті, що не виводяться з даної теорії) не завжди можна розпізнати. Це стосується теорій першого порядку, таких як арифметика Пеано. Однак для конкретної моделі, яка може бути описана логікою першого порядку, деякі твердження можуть бути істинними, але невирішуваними в межах теорії, що використовується для опису цієї моделі. Наприклад, згідно з теоремою Геделя про неповноту, ми знаємо, що в жодній несуперечливій теорії, аксіоми якої є істинними для натуральних чисел, не можна довести всі твердження логіки першого порядку, які є істинними для натуральних чисел, навіть якщо список аксіом дозволено зробити нескінченно обчислюваним. З цього випливає, що автоматизована система доведення теорем не завершить роботу під час пошуку доказу саме тоді, коли твердження, яке розглядається, є нерозв'язним у теорії, що використовується, навіть якщо воно істинне в моделі, яка нас цікавить.

Простіше завдання для подібних систем — це перевірка доказу, коли існуючий доказ теореми підтверджується як правильний або неправильний. Для цього вимагається, щоб кожен окремий крок доказу міг бути перевірений за допомогою примітивно-рекурсивної функції або програми, і, таким чином, ця проблема завжди є розв'язною. Це було доведено в роботі [19].

На сьогодні існують наступні системи: Lean 4, Coq, Isabelle/HOL [20, 21, 22]. Найкраще підходить для математики Lean 4 завдяки своїй математичній бібліотеці та проекту, який ставить собі на меті формалізувати основні концепції та теорії вищої математики [23, 24].

Нижче наведено код у Lean, який є теоремою з довденням ірраціональності $\sqrt{2}$:

```
theorem irrational_sqrt_two : Irrational (√2) := by
  simpa using Nat.prime_two.irrational_sqrt
```

Лістинг. 1. Приклад довдення теореми у Lean

Розберемо кожен оператор. irrational_sqrt_two — це назва теореми. Irrational (√2) означає, що квадратний корінь із 2 є ірраціональним числом. У Lean предикат Irrational означає, що число не може бути подане у вигляді відношення двох цілих чисел (тобто воно не є раціональним). simpa using — це тактика в Lean, яка спрощує поточну ціль шляхом переписування (rewriting) і застосування визначень. Вона часто використовується, щоб "очистити" доказ, коли результат майже очевидний. У цьому випадку simpa використовується для спрощення завдання, спираючись на вже відому теорему або факт Nat.prime_two.irrational_sqrt. Частина Nat.prime_two вказує на те, що число 2 є простим (prime) числом у математичній бібліотеці Lean, а irrational_sqrt — це властивість, пов'язана з простими числами у Lean. Вона говорить, що квадратний корінь з будь-якого простого числа є ірраціональним. Разом Nat.prime_two.irrational_sqrt — це теорема в Lean, яка конкретно стверджує, що √2 є ірраціональним, спираючись на загальну властивість квадратних коренів із простих чисел. Доказ спирається на загальний результат у бібліотеці Lean, що квадратний корінь із будь-якого простого числа є ірраціональним. Оскільки 2 є простим числом, теорема Nat.prime_two.irrational_sqrt безпосередньо застосовується. Тактика simpa спрощує поточну ціль доказу, використовуючи цю теорему.

В цьому розділі були розглянуті основні сучасні напрями реалізації моделей інтелектуального навчання математики, як для простих задач на застосування деякого алгоритму, так і складних – на складання алгоритму. Далі покажемо існуючі розробки – інтелектуальні комп’ютерні навчальні програми, що застосовують розглянуті практики.

## 4. Існуючі розробки

Розглянемо спочатку існуючі розробки для простих задач.

WeBWorK – це безкоштовна система на основі Perl для виконання індивідуальних домашніх завдань через Інтернет [25]. Спочатку він був розроблений у 1995 році професорами Арнольдом Пайзером і Майклом Гейджем на факультеті математики Рочестерського університету для використання в навчанні математики. Команда розробників із низки навчальних закладів тепер підтримує систему, яка зараз використовується для широкого спектру курсів з математики та суміжних дисциплін. WeBWorK (рис. 6) покращує освітній процес кількома способами. Надаючи учням негайний зворотний зв'язок щодо правильності їхніх відповідей, учні заохочуються робити кілька спроб, поки вони не досягнуть успіху. Індивідуалізуючи проблеми, обман не заохочується. Надаючи інструкторам статистичні дані в реальному часі, плани уроків можна налаштувати. Головна відмінність WeBWorK від інших веб-систем домашніх завдань полягає в тому, як написані завдання. Мова PG («генерування задач») дозволяє включати як код Perl, так і код LaTeX, дозволяючи авторам задач скористатися перевагами синтаксичної ефективності Perl і типографічної гнучкості LaTeX (що в основному необхідно для рендерингу математичних виразів). Концепція навчання простим задачам та задачам на застосування алгоритму успішно реалізована в даній системі.

WeBWorK використовує Apache з mod_perl, MySQL, LaTeX, dvipng, MathJax, графічну бібліотеку GD і багато модулів CPAN.

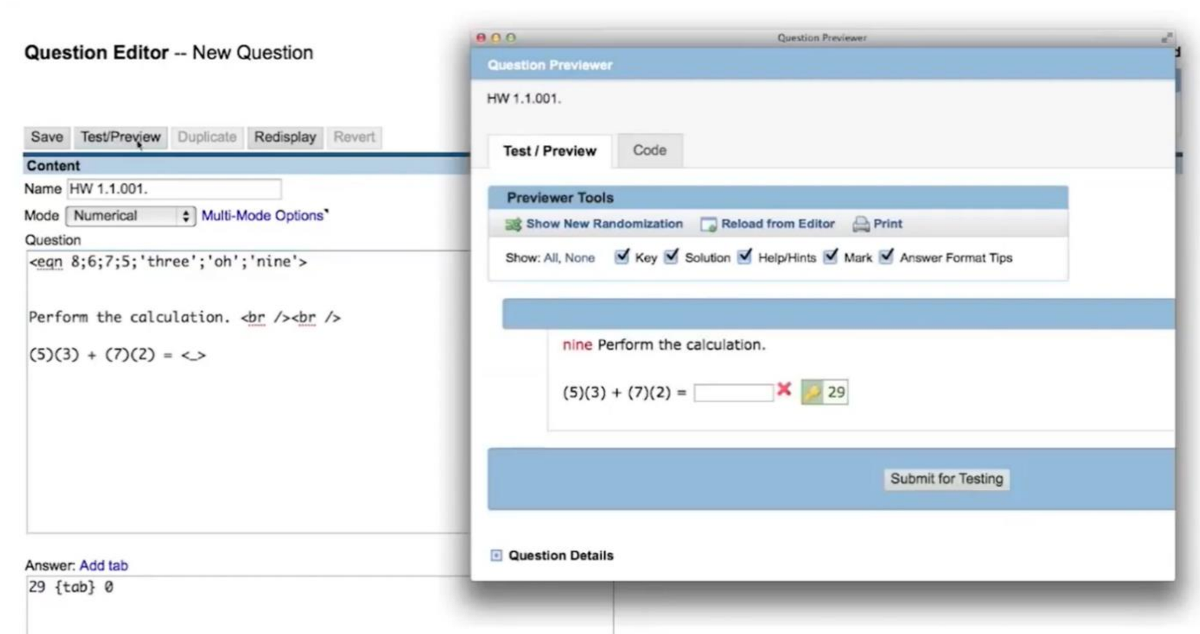



Рис. 6. WeBWorK

MyOpenMath – це онлайн-система (рис. 7) керування та оцінювання курсу з математики та інших кількісних галузей [26]. Основна увага MyOpenMath полягає в забезпеченні багатих алгоритмічно згенерованих оцінок для підтримки використання безкоштовних відкритих підручників. MyOpenMath.com надає керований хостинг програмного забезпечення онлайн-оцінювання IMathAS з відкритим кодом. Запитання та готові курси створюються викладачами в спільноті користувачів і діляться ними з іншими.

В свою чергу IMathAS – це Інтернет-система оцінювання з математики [27]. Запитання генеруються алгоритмом, а відповіді на числові та математичні вирази можна оцінювати комп'ютером. Окрім цього, IMathAS містить інструменти керування навчанням, зокрема розміщення оголошень, текстових файлів і вкладень, а також дискусійні форуми та повний журнал оцінок. У публікаціях і оцінках IMathAS дозволяє точно відображати математику та графіки з простим введенням математики у стилі калькулятора та створенням графіка за допомогою вказівника миші. Цей проект має зручний для користувача інтерфейс, але не всі аспекти інтелектуального навчання реалізовані, в тому числі і для простих задач.

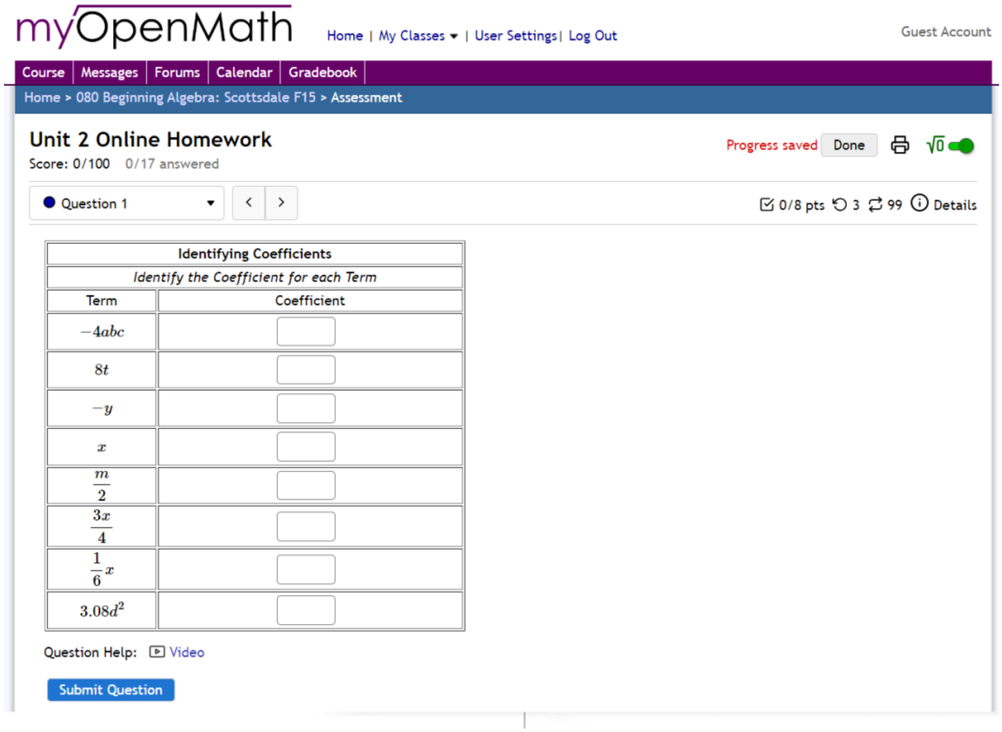



Рис. 7. MyOpenMath

WirisQuizzes [30] дозволяє викладачам створювати оцінювання STEM за допомогою рівнянь, графіків або текстових відповідей і автоматично виправляє відповіді студентів. Дозволяє динамічні запитання, додаючи випадкові параметри, таким чином підтримуючи запобігання шахрайству. WirisQuizzes легко інтегрується з різним робочим середовищем (рис. 8). Таким чином, Wiris Quizzes – це інструмент з комп’ютерним оцінюванням, який покращує звичайні стандартні типи запитань у різних курсах за допомогою математичних і наукових функцій та успішно реалізує модель навчання простим задачам.

MathType [31] – це редактор формул WYSIWYG (рис. 9) та математична нотація, створена WIRIS, яка інтегрована в курси. MathType дозволяє писати позначення (рівняння/формули/вирази тощо) від найпростіших до найскладніших, які можна використовувати для кожного зі своїх запитань.

WIRIS посилює ваші запитання, додаючи:

- випадкові величини (такі як поліноми, матриці, графіки);
- автоматичне оцінювання відповідей;
- графічні представлення в 2D і 3D;
- редактор формул для введення відповідей учнів;
- перевірка синтаксису відповіді для відкритих відповідей;
- відкриті запитання, як-от «Введіть дійсне число, яке не є раціональним» (є нескінченна кількість правильних і неправильних відповідей).

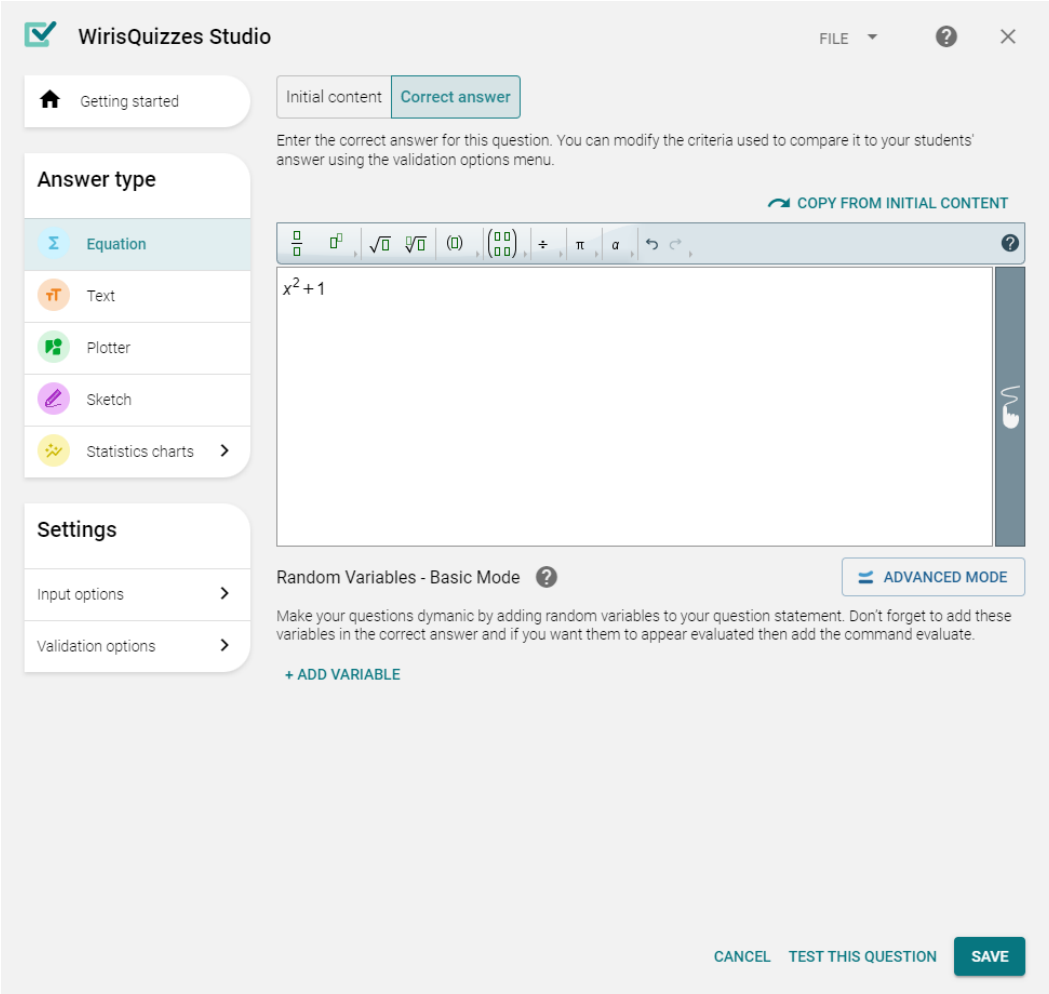



Рис. 8. WirisQuizzes Studio

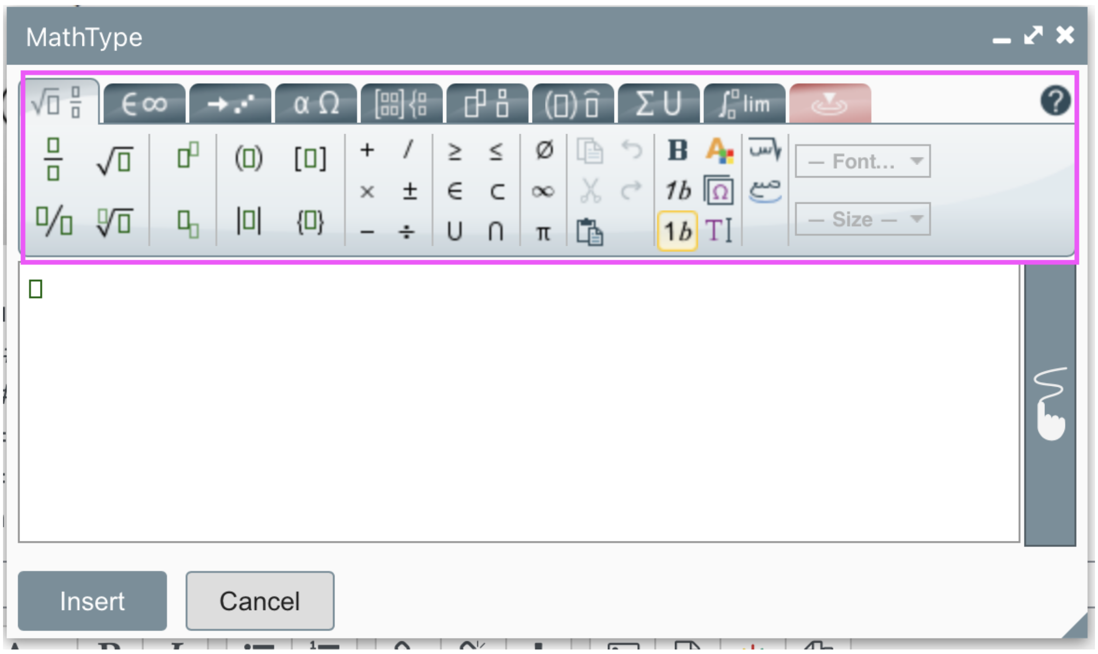



Рис. 9. Панель редактору MathType

WirisQuizzes успішно використовує всі можливості редактору MathType та навіть більше, оскільки містить зручні графічні елементи управління, що робить досвід користувача більш зручним.

За допомогою WIRIS можна генерувати кілька різних запитань з одним запитанням, встановлюючи випадкові значення для змінних у задачі. Кожного разу, коли запитання відображається, воно показує нові значення даних, хоча структура запитання однакова. Ці нові випадкові значення обчислюються онлайн у режимі реального часу.

Останнім часом з'являється багато навчальних ресурсів, у тому числі спрямованих на математику, з використанням сервісів генеративного штучного інтелекту, таких як, наприклад, ChatGPT. Дійсно, машинне навчання є потужним методом, і особливо за останнє десятиліття воно досягло неабияких успіхів, одним з останніх є і ChatGPT. Але без «справжнього розуміння математики» для ChatGPT практично неможливо надійно отримати правильну відповідь. Тим не менш, ChatGPT може навіть скласти дуже правдоподібне пояснення того, «як він отримав відповідь». Фундаментальна ідея системи штучного інтелекту на основі генеративної мови, як-от ChatGPT, просто не підходить у ситуаціях, коли потрібно робити структуровані обчислювальні роботи. Однак, є спосіб вирішити цю проблему – підключити ChatGPT до Wolfram|Alpha та всіх його «надздібностей» обчислювальних знань. Всередині Wolfram|Alpha все перетворюється на обчислювальну мову та точний код мови WolframAlpha [28]. З одного боку такий підхід має гнучкість у представленні користувачу певної задачі та можливих шляхів її вирішення. Однак недоліками такого підходу є неможливість впливати викладачем-асистентом на хід рішення задачі, та певна ймовірність, що складні задачі можуть бути розібрані з помилками. Для більшості простих задач ця система може бути корисна, але без 100% гарантії коректності. Тому її можна використовувати як додатковий інструмент підтримки навчання, але не як самодостатню навчальну систему, яка б покривала повний цикл навчання.

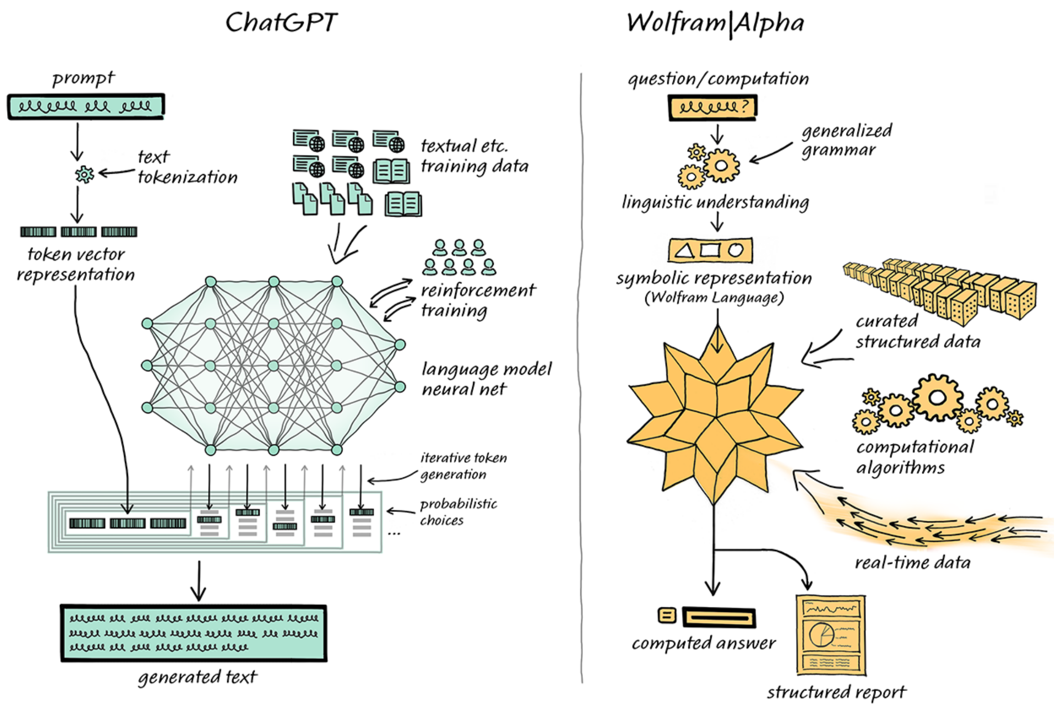



Рис. 10. Порівняння принципів ChatGPT та Wolfram|Alpha

Lean4 Natural Numbers Game – це ігрове середовище для навчання доведенню теорем в Lean. На рисунку 11 зображено використання середовища для доведення правила розкриття квадрату суми доданків. Досить зручне середовище, де навчальний курс побудований від навчання найпростіших концепцій до більш складних. Усі леми, аксіоми, правила та тактики доведення, які доступні студенту для їх застосування при доведенні основної теореми відкриваються по мірі освоєння нових правил та концепцій, від найпростіших до найскладніших [29]. Natural Number Game основана на lean4game технології, яку можна використовувати для розробки подібних навчальних ігор для навчання доведення теорем. Ідеально підійде для освоєння технік доведення та виводу формул, однак із недоліків – це те, що студенту необхідно вивчати синтаксис мови Lean.

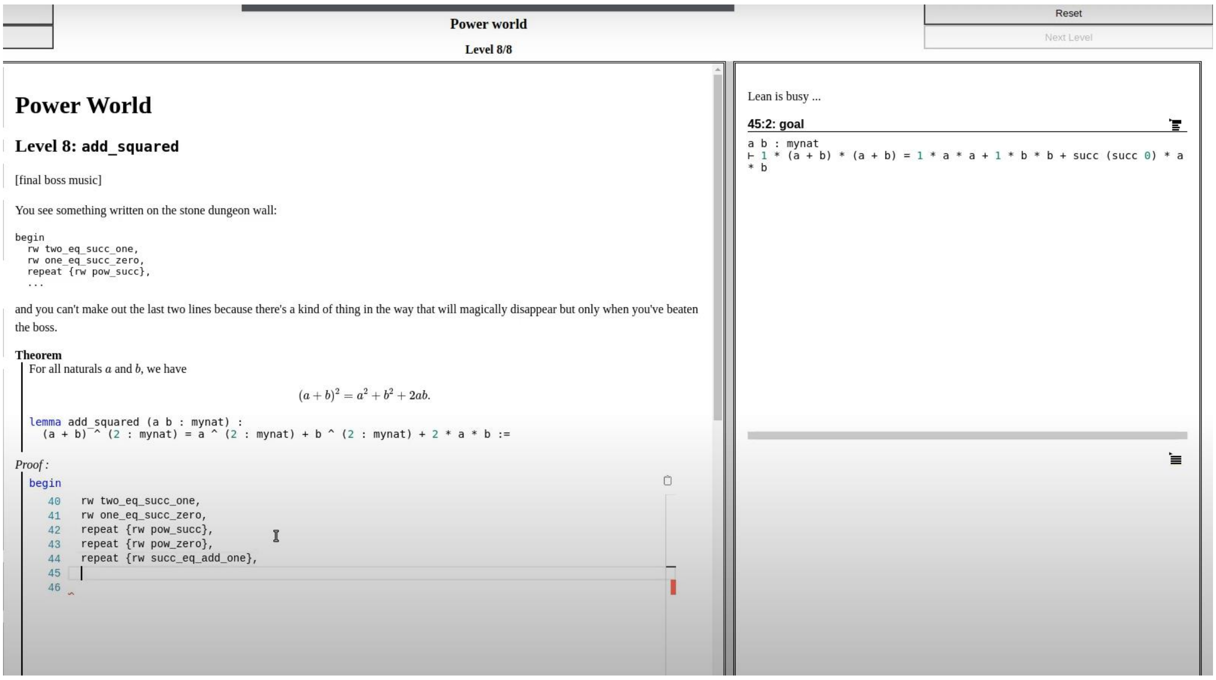



Рис. 11. Lean4 Natural Number Game

Порівняльна характеристика розробок наведена в таблиці 1.

Таблиця 1

Порівняльна характеристика існуючих розробок

| Назва\Підтримка | Прості задачі | Складні задачі | Зручний інтерфейс | Генерація тексту | Генерація точних значень | Доведення теорем |
|---|---|---|---|---|---|---|
| WeBWorK | + | - | +/- | - | + | - |
| MyOpenMath | + | - | + | - | - | - |
| WirisQuizzes | + | - | + | - | + | - |
| ChatGPT | +/- | - | - | + | +/- | +/- |
| WolframAlpha | +/- | - | - | + | +/- | +/- |
| Lean4 | - | + | - | - | +/- | + |

## 5. Наші розробки

Наша команда також розробляє власне рішення, яке враховує недоліки в існуючих проектах та пропонує кращі механізми інтелектуального навчання.

Проект Мирна, що розробляється та був розглянутий в роботі [2] включає в себе можливість навчання студентів вирішенню простих задач, складних задач, має зручний інтерфейс для користувача (як для студента, так і для викладача-асистента), використовує генеративний штучний інтелект для генерації різних варіантів постановки задачі, використовує систему генерації вхідних даних, відповідей та перевірки відповідей використовуючи Python, як мову програмування алгоритмів та LaTeX як мову розмітки для зберігання та відображення задачі. Також на етапі проектування знаходиться зручний графічний інтерфейс для системи перевірки доведення теорем та складних задач.

Екранна форма інтерфейсу при вирішенні простої задачі в форматі тесту, де пропонується обрати правильну відповідь із запропонованих варіантів, зображена на рис. 12.

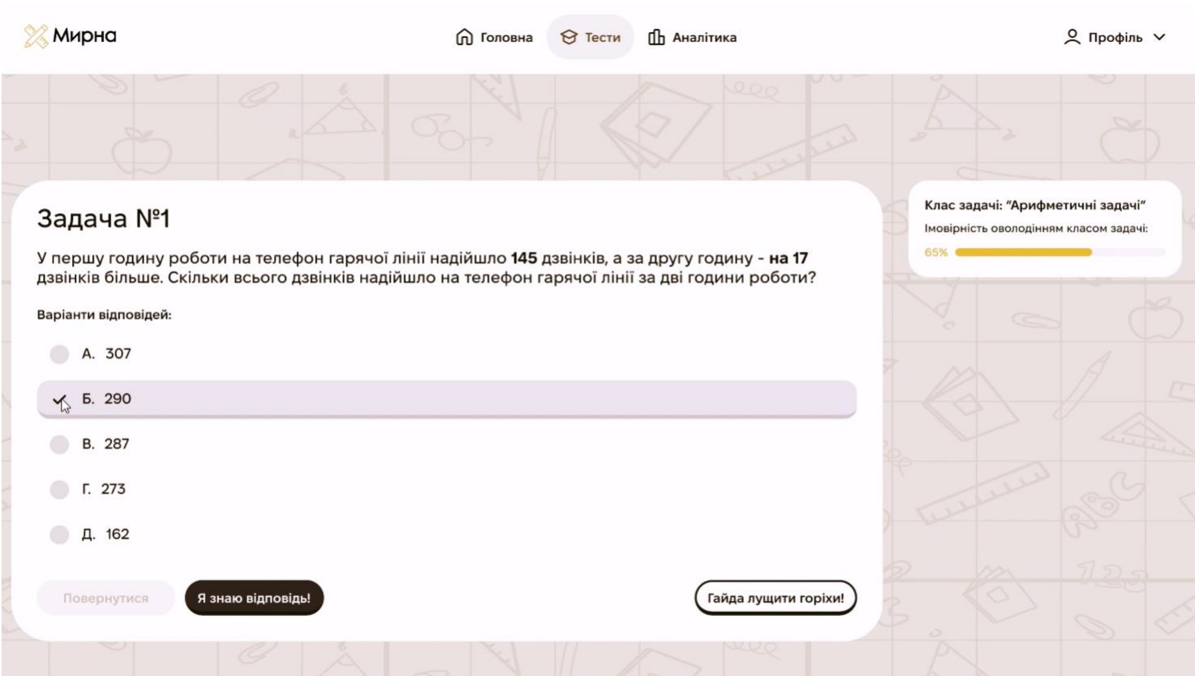



Рис. 12. ІКНП Мирна – проста задача в форматі тесту

Екранні форми інтерфейсу користувача при покроковому вирішенні простої задачі (рис. 13, 14). Учень використовує конструктор із «пазлів», в яких містяться правильні і неправильні значення. У випадку підстановки неправильного значення, система підсвічує його червоним та може надавати підказку, у випадку правильного – зеленим. Справа відстежується ймовірність оволодіння учнем класу задач. У випадку, коли учень надає неправильні відповіді, система знижує ймовірність оволодіння учнем компетенцій та знань необхідних для вирішення задач даного класу. І навпаки, коли учень надає правильні відповіді – збільшує цю ймовірність, допоки не досягне заданого порогу, коли можна вважати, що студент освоїв компетенції та знання необхідні для вирішення задач даного класу. Тоді ІКНП переводить учня до виконання задач наступного класу, де вивчаються інші математичні концепції.

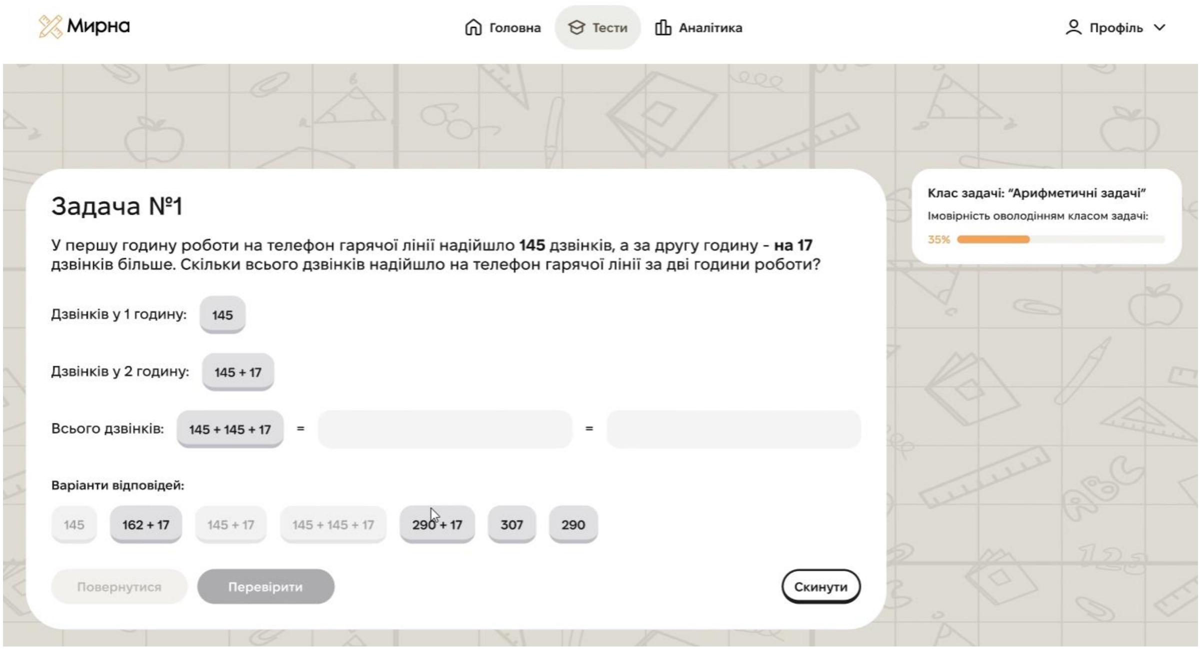



Рис. 13. ІКНП Мирна – рішення задачі в покроковому режимі

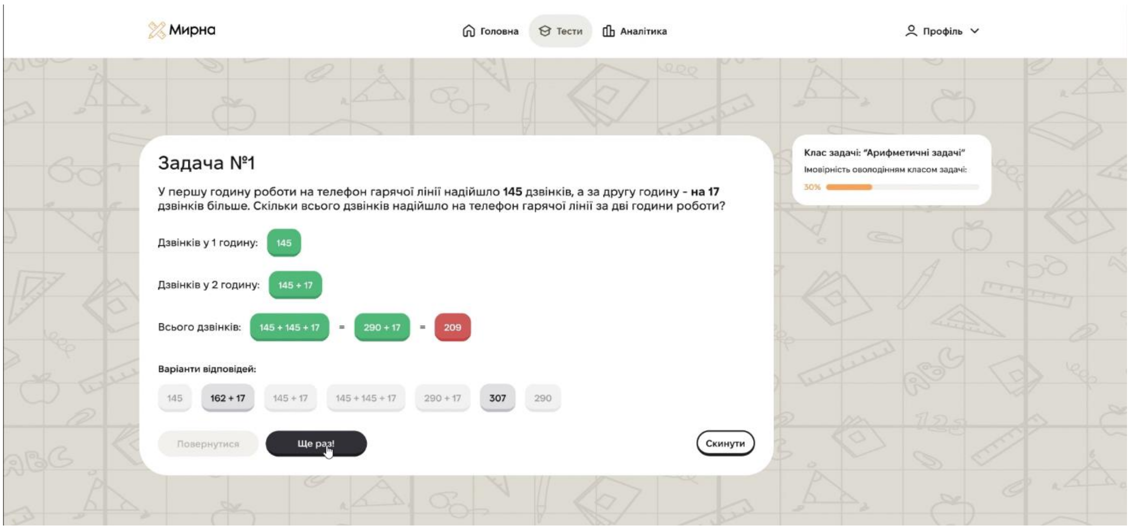



Рис. 14. ІКНП Мирна – підсвічування відповідей в покроковому режимі

Результат наданої неправильної відповіді на тестове завдання зображений на рис. 15. Для правильної адаптивної підказки в системі використовуються вбудовані діагностичні моделі, які заповнюються асистентом-викладачем.

Також в системі є рейтинг закладів освіти та інтелектуальні рекомендації, які можна застосувати для усунення негативних тенденцій в конкретному освітньому закладі, що зображено на екранних формах (рис. 16).

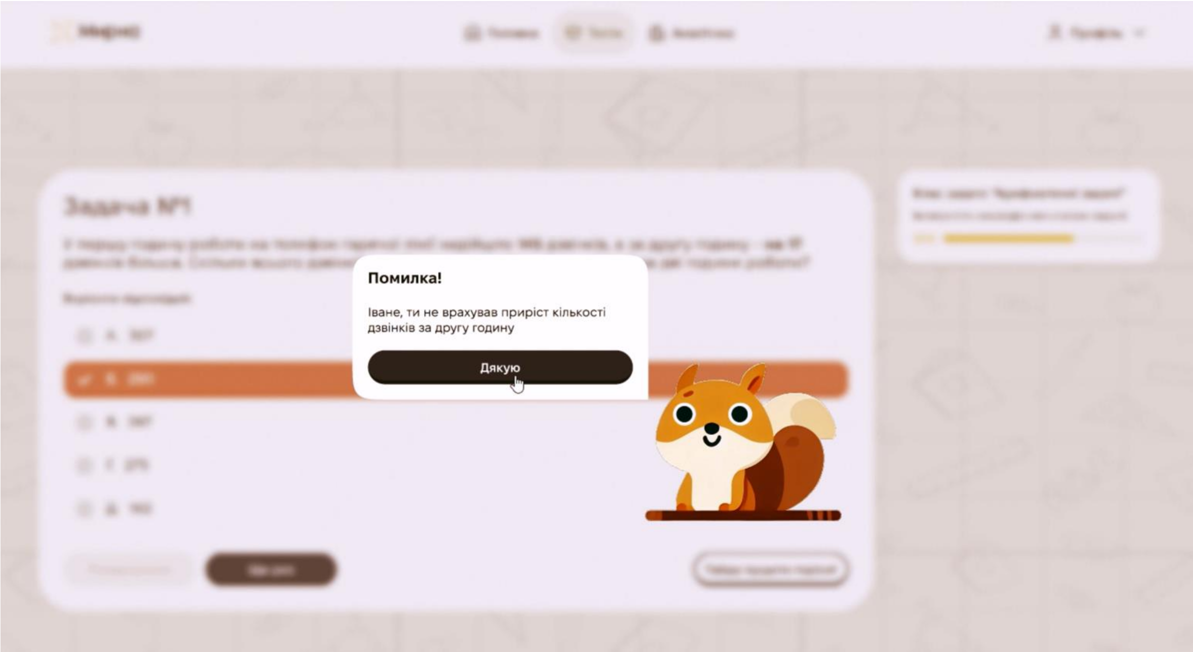
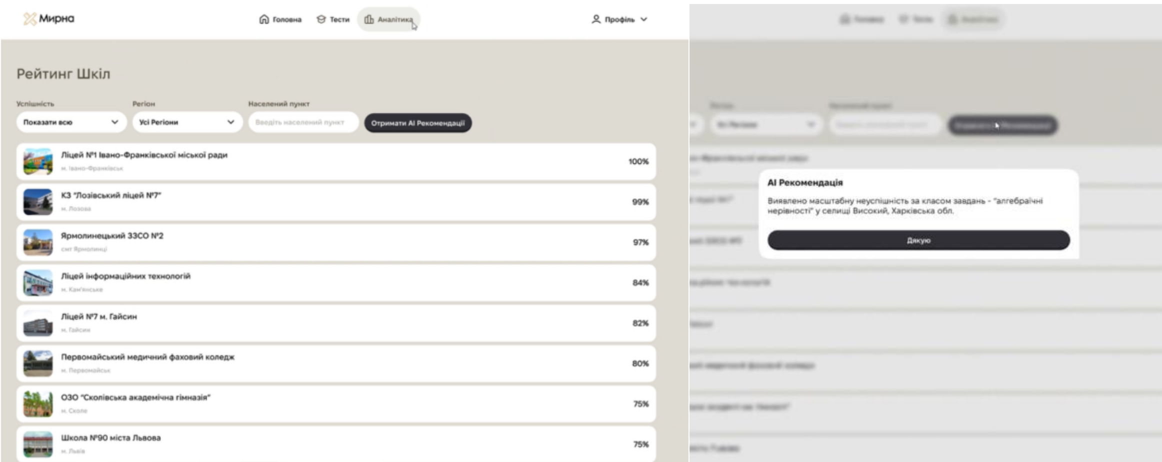



Рис. 15. ІКНП Мирна – адаптивна підказка




а б

Рис. 16. ІКНП Мирна – рейтинг закладів освіти:
а – список закладів з рейтингом, б – інтелектуальна AI рекомендація або погляд

## 6. Актуальні задачі та відкриті питання

Провівши поточне дослідження, було виявлено можливості та задачі для подальшого дослідження, а також відкриті питання, що пропонуються для дослідження читачам:

1. Дослідження зовнішнього циклу для запропонованої моделі, провести експерименти та знайти оптимальний підхід

2. Формалізація алгоритму генерації неправильних відповідей або проміжків відповідей $IsCorrect(ans) = False$, що повинні створювати можливість діагностувати пробіли знань в $R$. А саме $Diagnose(ans) = r \in R$.

3. Реалізація зручного графічного інтерфейсу на кшталт MathType для однієї із систем перевірки доведення теорем (Coq, Isabelle/HOL, Lean). Інтерфейс повинен складатись із зрозумілих викладачу-асистенту та студенту математичних символів, що б усувало необхідність вивчення синтаксису мови відповідної системи.

4. Проектування архітектури ІКНП та діагностичних моделей для адаптивного навчання рішенню складних задач на доведення теорем та складання алгоритму.

5. Оцінити час, що буде витрачатись викладачами для наповнення курсу складними задачами.

6. Дослідити ефективність мікронавчання для даної дисципліни, що відкриває можливість створення міні-ІКНП або систем підтримки навчання на основі носимих пристроїв (wearables) .

## 7. Висновок

В даній роботі були розглянуті найбільш перспективні напрямки для використання інтелектуальних навчальних систем (ITS) на курсах вищої математики. Основні виклики для реалізації подібної системи становлять технічна складність проєкту, необхідність залучення робочої команди професіоналів розробників програмного забезпечення та необхідність подальших досліджень в напрямку навчання рішенню складних задач.

Розробка та впровадження подібних систем в навчальний процес здатне збільшити мотивацію та ефективність навчання під час впливу сучасних існуючих негативних факторів.

Сподіваємось, що це дослідження стане в нагоді, як науковцям, викладачам так і керівникам стартапів сфери електронного навчання.

## 8. Примітка

Деякі структурні частини та частини тексту документа статті були згенеровані великою мовною моделлю ChatGPT-4o.

## Список літератури

1. Платон. Держава // пер. П. Шорі / Cambridge (Mass.): Harvard University Press, 1935. – Кн. VII

2. Розроблення і впровадження інтелектуальної комп’ютерної системи підготовки школярів України до НМТ з точних наук: звіт з НДР // А. Г. Чухрай та ін. – Харків : Нац. аерокосм. ун-т ім. М. Є. Жуковського «Харків. авіац. ін-т», 2024.

3. National Center for Education Statistics. *Mathematics Achievement of U.S. Students* / National Center for Education Statistics. – Washington, D.C.: U.S. Department of Education, 2023. – Режим доступу: https://nces.ed.gov/pubsearch/pubsinfo.asp?pubid=2024038

4. Ланда, Л. Н. Алгоритмізація в навчанні // Л. Н. Ланда / Просвітництво, 1966. – 523 с.

5. Кулік, А. С., Чухрай, А. Г., Гавриленко, О. В., Гайдачук, Д. О., Кулік, І. А. Інтелектуальна комп’ютерна підтримка навчання складанню алгоритмів та SQL-запитів: монографія // А. С. Кулік, А. Г. Чухрай, О. В. Гавриленко, Д. О. Гайдачук, І. А. Кулік / Харків : Нац. аерокосм. ун-т ім. М. Є. Жуковського «Харків. авіац. ін-т», 2020. – 192 с.

6. Чухрай, А. Г. Методологія навчання алгоритмам: монографія / А. Г. Чухрай / Харків : Нац. аерокосм. ун-т ім. М. Є. Жуковського «Харків. авіац. ін-т», 2017. – 336 с.

7. Крістенсен, М., Ларсен, Д., Сейделін, Л., Свобо, К. Роль математики в STEM-активностях: синтези та фреймворк з огляду літератури / М. Крістенсен, Д. Ларсен, Л. Сейделін, К. Свобо // International Journal of Education in Mathematics, Science and Technology. – 2023. – Т. 12. – с. 418–431.

8. Український центр оцінювання якості освіти. Звіт про результати основної сесії національного мультипредметного тесту 2023 року. Том 1 / Український центр оцінювання якості освіти. – Київ : УЦОЯО, 2023. – 116 с. – Режим доступу: https://testportal.gov.ua//wp-content/uploads/2023/08/ZVIT-NMT_2023-Tom_1_.pdf

9. Український центр оцінювання якості освіти. Звіт про результати основної сесії національного мультипредметного тесту 2024 року. Том 1 / Український центр оцінювання якості освіти. – Київ : УЦОЯО, 2024. – 148 с. – Режим доступу: https://testportal.gov.ua/wp-content/uploads/2024/10/Zvit_NMT_2024_Tom_I_gotovyj_onovlenyj.pdf

10. Андріс, М., Віллемс, К., Фійє, Б., Де Волф, П. Нейрокогнітивний механізм, що лежить в основі уникнення математики серед людей, які відчувають математичну тривогу / М. Андріс, К. Віллемс, Б. Фійє, П. Де Волф // ResearchGate. – 2023.

11. OECD. Математика для життя і роботи: порівняльна перспектива щодо математики для інформування реформи старшої школи в Англії / OECD Publishing. – Париж, 2024. – Доступно за посиланням: https://doi.org/10.1787/26f18d39-en.

12. OECD. Стан навчання та рівності в освіті: Міжнародний звіт за результатами міжнародного дослідження якості освіти PISA-2022. Том I / Пер. з англ. – Київ: Український центр оцінювання якості освіти, 2024. – 520 с.

13. Ярема, А. І. Феномен успіху в соціологічному вимірі / А. І. Ярема / Львів: Національний університет “Львівська політехніка” / ВІСНИК ЛЬВІВ. УН-ТУ / Серія соціол. 2010. Вип. 4. С. 92–99

14. OECD,. (2010). The High Cost of Low Educational Performance: The Long-run Economic Impact of Improving PISA Outcomes.

15. Vanlehn, K. The behavior of tutoring systems [Text] / K. Vanlehn // Artificial intelligence in education. – 2006. – Vol. 16, № 3. – P. 227-265.

16. Shailendra, Samar & Kadel, Rajan & Sharma, Aakanksha. Framework for Adoption of Generative Artificial Intelligence (GenAI) in Education [Text] // Shailendra, Samar & Kadel, Rajan & Sharma, Aakanksha // Австралія: SITE, Мельбурнський технологічний інститут, 2024.

17. Gödel, K. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme [Text] / K. Gödel // Monatshefte für Mathematik und Physik. – 1931. – Vol. 38. – P. 173-198.

18. Church, A. An unsolvable problem of elementary number theory [Text] / A. Church // American Journal of Mathematics. – 1936. – Vol. 58, № 2. – P. 345–363.

19. Kleene, S. C. Introdu*ction to Metamathematics* [Text] / S. C. Kleene // Amsterdam: North-Holland Publishing Company, 1971. – 550 p.

20. De Moura, L. M., Avigad, J., Kong, S., & Roux, C. *The Lean 4 Theorem Prover and Programming Language (System Description)* [Text] / L. M. de Moura, J. Avigad, S. Kong, C. Roux // Proceedings of the 13th International Conference on

Automated Deduction (CADE-27). – 2021. – P. 625–635.

21. Paulin-Mohring, C. Introduction to the Coq Proof-Assistant for Practical Software Verification [Text] / C. Paulin-Mohring // Workshop on Learning from Authoritative Security Experiment Results. – 2011. – Режим доступу: https://api.semanticscholar.org/CorpusID:17057477

22. Blanchette, J. Automatic Proof and Disproof in Isabelle/HOL [Text] / J. Blanchette, L. Bulwahn, T. Nipkow // Lecture Notes in Computer Science. – 2011. – Vol. 6890. – P. 12–27. – DOI: 10.1007/978-3-642-24364-6_2.

23. Buzzard, K. What is the Xena Project? [Text] / K. Buzzard // Xena: A Project at Imperial College London. – 2019. – Режим доступу: https://www.ma.imperial.ac.uk /~buzzard/xena/ (дата звернення: 29 листопада 2024).

24. Avigad, J., & Massot, P. *Mathematics in Lean. Release 0.1* [Text] / J. Avigad, P. Massot. – Nov 11, 2024. – Режим доступу: https://leanprover-community.github.io/mathematics_in_lean/mathematics_in_lean.pdf (дата звернення: 29 листопада, 2024).

25. Головна сторінка WeBWorK [Електронний ресурс]. – Режим доступу: https://webwork.maa.org/wiki/WeBWorK_Main_Page (дата звернення: 29 листопада 2024 року).

26. MyOpenMath [Електронний ресурс]. – Режим доступу: https://www.myopenmath.com/index.php (дата звернення: 29 листопада 2024 року).

27. IMathAS [Електронний ресурс]. – Режим доступу: https://www.imathas.com/ (дата звернення: 29 листопада 2024 року).

28. Wolfram, S. Wolfram|Alpha as the Way to Bring Computational Knowledge Superpowers to ChatGPT [Електронний ресурс] / S. Wolfram. – 2023. – Режим доступу: https://writings.stephenwolfram.com/2023/01/wolframalpha-as-the-way-to-bring-computational-knowledge-superpowers-to-chatgpt/ (дата звернення: 29 листопада 2024 року).

29. Lean Game Server [Електронний ресурс]. – Режим доступу: https://adam.math.hhu.de/ (дата звернення: 29 листопада 2024 року).

30. WIRIS Quizzes Demo [Електронний ресурс]. – Режим доступу: https://www.wiris.com/wirisquizzes/demo/ (дата звернення: 29 листопада 2024 року).

31. MathType by WIRIS [Електронний ресурс]. – Режим доступу:https://www.wiris.com/en/mathtype/ (дата звернення: 29 листопада 2024 року).

## References

1. Platon. Derzhava // per. P. Shori / Cambridge (Mass.): Harvard University Press, 1935. – Kn. VII

2. Chuxraj, A. G., Kulik A. S., Zelenyak O. P., Proxorov O. V., ta inshi // Rozroblennya i vprovadzhennya intelektual`noyi komp'yuternoyi sy`stemy` pidgotovky` shkolyariv Ukrayiny` do NMT z tochny`x nauk: zvit z NDR / A. G. Chuxraj, A. S. Kulik, O. P. Zelenyak, O. V. Proxorov, ta inshi / Xarkiv : Nacz. aerokosm. un-t im. M. Ye. Zhukovs`kogo «Xarkiv. aviacz. in-t», 2024.

3. National Center for Education Statistics. Mathematics Achievement of U.S. Students / National Center for Education Statistics. – Washington, D.C.: U.S. Department of Education, 2023. – Rezhy`m dostupu: https://nces.ed.gov/pubsearch/pubsinfo.asp?pubid=2024038

4. Landa, L. N. Algory`tmizaciya v navchanni / L. N. Landa / Prosvitny`cztvo,

1966. – 523 s.

5. Kulik, A. S., Chuxraj, A. G., Gavry`lenko, O. V., Gajdachuk, D. O., Kulik, I. A. Intelektual`na komp'yuterna pidtry`mka navchannya skladannyu algory`tmiv ta SQL-zapy`tiv: monografiya // A. S. Kulik, A. G. Chuxraj, O. V. Gavry`lenko, D. O. Gajdachuk, I. A. Kulik / Xarkiv : Nacz. aerokosm. un-t im. M. Ye. Zhukovs`kogo «Xarkiv. aviacz. in-t», 2020. – 192 s.

6. Chuxraj, A. G. Metodologiya navchannya algory`tmam: monografiya // A. G. Chuxraj / Xarkiv : Nacz. aerokosm. un-t im. M. Ye. Zhukovs`kogo «Xarkiv. aviacz. in-t», 2017. – 336 s.

7. Kristensen, M., Larsen, D., Sejdelin, L., Svobo, K. Rol` matematy`ky` v STEM-akty`vnostyax: sy`ntezy` ta frejmvork z oglyadu literatury` / M. Kristensen, D. Larsen, L. Sejdelin, K. Svobo // International Journal of Education in Mathematics, Science and Technology. – 2023. – T. 12. – s. 418–431.

8. Ukrayins`ky`j centr ocinyuvannya yakosti osvity`. Zvit pro rezul`taty` osnovnoyi sesiyi nacional`nogo mul`ty`predmetnogo testu 2023 roku. Tom 1 / Ukrayins`ky`j centr ocinyuvannya yakosti osvity`. – Ky`yiv : UCzOYaO, 2023. – 116 s. – Rezhy`m dostupu: https://testportal.gov.ua//wp-content/uploads/2023/08/ZVIT-NMT_2023-Tom_1_.pdf

9. Ukrayins`ky`j centr ocinyuvannya yakosti osvity`. Zvit pro rezul`taty` osnovnoyi sesiyi nacional`nogo mul`ty`predmetnogo testu 2024 roku. Tom 1 / Ukrayins`ky`j centr ocinyuvannya yakosti osvity`. – Ky`yiv : UCzOYaO, 2024. – 148 s. – Rezhy`m dostupu: https://testportal.gov.ua/wp-content/uploads/2024/10/Zvit_NMT_2024_Tom_I_gotovyj_onovlenyj.pdf

10. Andris, M., Villems, K., Fijye, B., De Volf, P. Nejrokognity`vny`j mexanizm, shho lezhy`t` v osnovi uny`knennya matematy`ky` sered lyudej, yaki vidchuvayut` matematy`chnu try`vogu / M. Andris, K. Villems, B. Fijye, P. De Volf // ResearchGate. – 2023.

11. OECD. Matematy`ka dlya zhy`ttya i roboty`: porivnyal`na perspekty`va shhodo matematy`ky` dlya informuvannya reformy` starshoyi shkoly` v Angliyi / OECD Publishing. – Pary`zh, 2024. – Dostupno za posy`lannyam: https://doi.org/10.1787/26f18d39-en.

12. OECD. Stan navchannya ta rivnosti v osviti: Mizhnarodny`j zvit za rezul`tatamy` mizhnarodnogo doslidzhennya yakosti osvity` PISA-2022. Tom I / Per. z angl. – Ky`yiv: Ukrayins`ky`j centr ocinyuvannya yakosti osvity`, 2024. – 520 s.

13. Yarema, A. I. Fenomen uspixu v sociologichnomu vy`miri / A. I. Yarema / L`viv: Nacional`ny`j universy`tet "L`vivs`ka politexnika" / VISNY`K L`VIV. UN-TU / Seriya sociol. 2010. Vy`p. 4. S. 92–99

14. OECD,. (2010). The High Cost of Low Educational Performance: The Long-run Economic Impact of Improving PISA Outcomes.

15. Vanlehn, K. The behavior of tutoring systems [Text] / K. Vanlehn // Artificial intelligence in education. – 2006. – Vol. 16, # 3. – P. 227-265.

16. Shailendra, Samar & Kadel, Rajan & Sharma, Aakanksha. Framework for Adoption of Generative Artificial Intelligence (GenAI) in Education [Text] // Shailendra, Samar & Kadel, Rajan & Sharma, Aakanksha // Avstraliya: SITE, Mel`burns`ky`j texnologichny`j insty`tut, 2024.

17. Gödel, K. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme [Text] / K. Gödel // Monatshefte für Mathematik und Physik. – 1931. – Vol. 38. – R. 173-198.

18. Church, A. An unsolvable problem of elementary number theory [Text] /

A. Church // American Journal of Mathematics. – 1936. – Vol. 58, # 2. – R. 345–363.
19. Kleene, S. C. Introduction to Metamathematics [Text] / S. C. Kleene // Amsterdam: North-Holland Publishing Company, 1971. – 550 p.
20. De Moura, L. M., Avigad, J., Kong, S., & Roux, C. The Lean 4 Theorem Prover and Programming Language (System Description) [Text] / L. M. de Moura, J. Avigad, S. Kong, C. Roux // Proceedings of the 13th International Conference on Automated Deduction (CADE-27). – 2021. – P. 625–635.
21. Paulin-Mohring, C. Introduction to the Coq Proof-Assistant for Practical Software Verification [Text] / C. Paulin-Mohring // Workshop on Learning from Authoritative Security Experiment Results. – 2011. – Rezhy`m dostupu: https://api.semanticscholar.org/CorpusID:17057477
22. Blanchette, J., Bulwahn, L., & Nipkow, T. Automatic Proof and Disproof in Isabelle/HOL [Text] / J. Blanchette, L. Bulwahn, T. Nipkow // Lecture Notes in Computer Science. – 2011. – Vol. 6890. – P. 12–27. – DOI: 10.1007/978-3-642-24364-62.
23. Buzzard, K. What is the Xena Project? [Text] / K. Buzzard // Xena: A Project at Imperial College London. – 2019. – Rezhy`m dostupu: https://www.ma.imperial.ac.uk/~buzzard/xena/ (data zvernennya: 29 ly`stopada 2024).
24. Avigad, J., & Massot, P. Mathematics in Lean. Release 0.1 [Text] / J. Avigad, P. Massot. – Nov 11, 2024. – Rezhy`m dostupu: https://leanprover-community.github.io/mathematics_in_lean/mathematics_in_lean.pdf (data zvernennya: 29 ly`stopada, 2024).
25. Golovna storinka WeBWorK [Elektronny`j resurs]. – Rezhy`m dostupu: https://webwork.maa.org/wiki/WeBWorK_Main_Page (data zvernennya: 29 ly`stopada 2024 roku).
26. MyOpenMath [Elektronny`j resurs]. – Rezhy`m dostupu: https://www.myopenmath.com/index.php (data zvernennya: 29 ly`stopada 2024 roku).
27. IMathAS [Elektronny`j resurs]. – Rezhy`m dostupu: https://www.imathas.com/ (data zvernennya: 29 ly`stopada 2024 roku).
28. Wolfram, S. Wolfram|Alpha as the Way to Bring Computational Knowledge Superpowers to ChatGPT [Elektronny`j resurs] / S. Wolfram. – 2023. – Rezhy`m dostupu: https://writings.stephenwolfram.com/2023/01/wolframalpha-as-the-way-to-bring-computational-knowledge-superpowers-to-chatgpt/ (data zvernennya: 29 ly`stopada 2024 roku).
29. Lean Game Server [Elektronny`j resurs]. – Rezhy`m dostupu: https://adam.math.hhu.de/ (data zvernennya: 29 ly`stopada 2024 roku).
30. WIRIS Quizzes Demo [Elektronny`j resurs]. – Rezhy`m dostupu: https://www.wiris.com/wirisquizzes/demo/ (data zvernennya: 29 ly`stopada 2024 roku).
31. MathType by WIRIS [Elektronny`j resurs]. – Rezhy`m dostupu: https://www.wiris.com/en/mathtype/ (data zvernennya: 29 ly`stopada 2024 roku).

Надійшла в редакцію 18.12.2024, розглянута на редколегії 18.12.2024

## Possibilities of using Intelligent Tutoring Systems (ITS) in higher mathematics courses

The current state of mathematical knowledge among school graduates in Ukraine and worldwide is analyzed. A negative trend has been identified, requiring urgent changes in educational approaches. This study explores the potential of using intelligent computer-based learning programs to mitigate the adverse effects on the quality of mathematical education. Existing models of such programs are analyzed, and their application to mathematics teaching is examined. Mathematical modeling and analysis of open-source information were used as research methods.

The results of the study include the development of mathematical models for intelligent computer-based mathematics learning. A classification of tasks into two subgroups is proposed: simple tasks (focused on algorithm application) and complex tasks (focused on algorithm creation). Key differences in the implementation of learning models for each class of tasks are examined. Particular attention is paid to tasks involving algorithm creation, such as theorem proving and formula derivation. A novel approach to teaching the resolution of such tasks is proposed, considering their unique structure and complexity.

Additionally, the potential of using generative artificial intelligence to automate the creation of unique tasks and databases is explored. A combination of algorithmic precision and lexical flexibility of generative AI models is proposed to build comprehensive models for any educational task. Existing projects and technologies in this field are analyzed, with a comparative assessment of their advantages and disadvantages. A promising and underexplored area is identified—the development of a mathematical user interface for automated theorem-proof verification systems, which is crucial for improving the quality of mathematical education.

The results of the study open new opportunities for enhancing intelligent learning systems and have practical value for engineers and entrepreneurs. The work provides researchers with a foundation for further investigations in the field of mathematical education automation and its adaptation to modern challenges.

***Key words***: CT, ICTP, PISA; ITS; AI; ITP; LLM; proof verification system; metamathematics.

**Відомості про авторів:**

**Чухрай Андрій Григорович** – доктор технічних наук, професор, Національний аерокосмічний університет ім. М. Є. Жуковського «Харківський авіаційний інститут». Електронна пошта: achukhray@gmail.com ORCID: 0000-0002-8075-3664.

**Столяренко Тетяна Леонідівна** – кандидат педагогічних наук, старший викладач, Відокремлений структурний підрозділ «Харківський фаховий коледж інформаційних технологій Національного аерокосмічного університету ім. М. Є. Жуковського «Харківський авіаційний інститут». Електронна пошта: t_stol@ukr.net ORCID: 0000-0002-7202-316X.

**Євдокимов Олександр Олегович** – магістр, аспірант, Національний аерокосмічний університет ім. М. Є. Жуковського «Харківський авіаційний інститут». Електронна пошта: o.yevdokymov@khai.edu ORCID: 0009-0008-9687-6344.

**Дем'яненко Владислав Анатолійович** – магістр, старший викладач, Національний аерокосмічний університет ім. М. Є. Жуковського «Харківський авіаційний інститут». Електронна пошта: v.demianenko@khai.edu ORCID: 0000-0002-1078-0114.

**About the Authors:**

**Chukhray Andriy Grygorovych** – Doctor of Science, Professor, M. Ye. Zhukovsky National Aerospace University “Kharkiv Aviation Institute”. E-mail: achukhray@gmail.com ORCID: 0000-0002-8075-3664.

**Stoliarenko Tetiana Leonidivna** – candidate of Pedagogical Sciences, Senior Lecturer, Separate Structural Unit “Kharkiv Professional College of Information Technologies of the M. Ye. Zhukovsky National Aerospace University “Kharkiv Aviation Institute”. E-mail: t_stol@ukr.net ORCID: 0000-0002-7202-316X.

**Yevdokymov Oleksandr Olegovych** – Master, PhD Student, M. Ye. Zhukovsky National Aerospace University “Kharkiv Aviation Institute”. E-mail: o.yevdokymov@khai.edu ORCID: 0009-0008-9687-6344.

**Dem'yanenko Vladyslav Anatolijovych** – Master, Senior Lecturer, M. Ye. Zhukovsky National Aerospace University “Kharkiv Aviation Institute”. E-mail: v.demianenko@khai.edu ORCID: 0000-0002-1078-0114.