

Purpose of this article is to study the effectiveness of classical and heuristic task scheduling algorithms in cloud environments based on simulation modelling and to develop a comprehensive methodology for their quantitative evaluation.

Results. A modular Python-based software complex was developed to conduct simulation experiments at three load levels: Low Load (10 tasks), Medium Load (20 tasks), and High Load (40 tasks) across clusters of 2 to 6 servers. Five algorithms were compared: FCFS, Round Robin, LPT, SPT, and Weighted Round Robin. A composite efficiency metric, Score, was proposed to jointly evaluate schedule quality (Makespan) and computational overhead of the scheduling algorithm itself. Experimental results confirm that the LPT algorithm achieves the best Makespan values under medium and high load conditions, while WRR delivers the best integral Score in heterogeneous server configurations. No universal optimal algorithm was identified.

Conclusion. The choice of scheduling algorithm must depend on system priorities: LPT is recommended for batch processing workloads where minimising total execution time is critical; WRR – for heterogeneous infrastructures where balanced resource utilisation is paramount; Round Robin and FCFS – for lightweight real-time scenarios with strict latency constraints. The proposed Score metric provides a practical tool for adaptive algorithm selection in Cloud Management Systems.

Keywords: cloud computing, resource scheduling, task distribution algorithms, Makespan, load balancing, LPT, Round Robin, WRR, integral efficiency metric.

Одержано редакцією 04.11.2025 р.
Прийнято до публікації 17.12.2025 р.

УДК 004.42:004.421

DOI 10.31651/2076-5886-2025-1-32-45

PACS89.20.Ff, 02.70.Wz

ЛИСЕНКО Олександр Володимирович
студент спеціальності «Прикладна фізика та наноматеріали» Черкаського національного університету імені Богдана Хмельницького
e-mail:
lysenko.oleksandr1623@vu.cdu.edu.ua

ТАТАРЧУК Євгеній Вікторович
кандидат фізико-математичних наук,
доцент кафедри фізики Черкаського національного університету імені Богдана Хмельницького
e-mail: Tatarchuk@vu.cdu.edu.ua

РОЗРОБКА ЗАСТОСУНКУ З МЕТОЮ КОГНІТИВНОГО ТРЕНУВАННЯ ТА АЛГОРИТМУ ДЛЯ АВТОМАТИЧНОЇ ГЕНЕРАЦІЇ АЛГЕБРАЇЧНИХ РІВНОСТЕЙ

У статті розглянуто проблему зниження когнітивних функцій як у літніх людей, так і у молодого покоління, що пов'язано зі зменшенням навичок усного рахування на тлі зростаючої залежності від цифрових технологій. Проаналізовано існуючі ігрові рішення для когнітивної стимуляції (Math Puzzle, Math Fight, Mathemagics), визначено їх переваги та недоліки. На основі проведеного огляду запропоновано власний підхід до створення веб-застосунку для розвитку когнітивних здібностей, який реалізує автоматичну генерацію алгебраїчних рівностей із використанням власного алгоритму розбору виразів на основі абстрактного синтаксичного дерева (AST). Розроблена бібліотека дозволяє ефективно перевіряти алгебраїчні рівності з мінімальним споживанням оперативної пам'яті порівняно з вбудованою функцією eval у Python. Представлено архітектуру веб-застосунку, REST API та результати порівняльного аналізу продуктивності.

Ключові слова: когнітивна стимуляція, веб-застосунок, токен, рекурсія, AST, API.

Вступ

У сучасному суспільстві дедалі більшої актуальності набуває проблема підтримки когнітивних функцій людини. З одного боку, люди похилого віку стикаються з поступовим зниженням пам'яті та швидкості мислення, що негативно позначається на їх повсякденній діяльності. З іншого боку, серед молоді спостерігається тенденція до втрати навичок усного рахування через надмірне використання калькуляторів і цифрових технологій. Це створює необхідність у пошуку нових методів тренування інтелектуальних здібностей, зокрема у формі інтерактивних ігор, які поєднують навчальний і розважальний аспекти.

Відомі приклади ігор для когнітивної стимуляції (Math Puzzle, Math Fight, Mathemagics) демонструють потенціал такого підходу, проте мають низку обмежень: відсутність веб-версій, перевантажений інтерфейс або жорсткі платні умови використання. Це знижує доступність і зручність їх застосування для широкого кола користувачів.

Метою даної роботи є розробка власного веб-застосунку для когнітивної стимуляції, який базується на алгоритмі генерації алгебраїчних рівностей із застосуванням парсера на основі абстрактного синтаксичного дерева. Запропонована бібліотека забезпечує стабільну роботу з великими масивами рівностей, мінімізує споживання оперативної пам'яті та надає користувачеві простий у використанні інтерфейс без реклами.

Мета статті – провести аналіз деяких сучасних ігор для розвитку когнітивної стимуляції, розробити власну гру та описати власну бібліотеку.

Виклад основного матеріалу:

1. Когнітивні функції

У сучасному світі люди в похилому віці починають втрачати пам'ять, що призводить до зменшення когнітивних функцій. Водночас зі стрімким розвитком ІТ - індустрії молодь поступово втрачає навички усного рахування, внаслідок цього люди починають бути залежні від технологій [1].

Когнітивні функції мозку – це здатність розуміти, пізнавати, вивчати, усвідомлювати, сприймати і переробляти зовнішню інформацію.

2. Класифікація когнітивних функцій

Сприйняття (гнозис) – здатність сприймати та розпізнавати інформацію, що надходить від органів чуття;

Пам'ять – здатність запам'ятовувати, зберігати та багаторазово відтворювати отриману інформацію;

Психомоторна функція (пракис) – здатність складати, зберігати та виконувати рухові програми;

Мова та мовлення – здатність до вербальної комунікації, включаючи розуміння зверненої мови, побудова власного мовного висловлювання, читання та письмо;

Увага – здатність своєчасно реагувати на сигнали, що надходять з органів чуття, концентруватися і зберігати протягом необхідного часу розумову працездатність, розділяти інформаційні потоки;

Керуючі функції – здатність до планування та контролю пізнавальної діяльності та поведінки, включаючи вибір мети (цілепокладання), побудова програми (програмування), перехід з одного етапу програми на інший (переключення, інтелектуальна адаптація) та зіставлення отриманого результату з метою (контроль);

Соціальний інтелект – здатність до розуміння емоцій та логіки інших людей.

Замість того, щоб люди проходили курси по розвитку когнітивних функцій, можна дати людям гру для розвитку когнітивної зони і не витратити час на повторне навчання [2]. Розробці подібній системі, як раз і присвячена дана робота.

Послаблення когнітивних функцій у людей можна визначити, як погіршення навичок усного рахування, як у похилому віці так і у молоді.

3. Огляд ігор для когнітивної стимуляції

Щоб створити власну гру для тренування когнітивної стимуляції, потрібно дослідити відомі ігри для того, щоб мати представлення, як виглядають такі ігри, та дослідити їх переваги й недоліки. На основі отриманих результатів спроектувати власну гру з уникненням типових помилок. Для цього дослідимо такі ігри:

- Math Puzzle;
- Math Fight;
- Mathemagics.

Math Puzzle – це інтелектуальна гра, призначена для розвитку та підтримки когнітивних функцій у користувачів різного віку. Вона була створена з метою тренування навичок усного рахування та логічного мислення в умовах, коли технології дедалі більше заміщують ці вміння у повсякденному житті [3]. Гра побудована на принципі складання правильних алгебраїчних рівностей, використовуючи задані числа та математичні операції (+, −, ×, ÷) (рис. 1.).

Алгоритм роботи гри полягає у випадковій генерації математичних завдань відповідно до обраного рівня складності. Система автоматично перевіряє правильність введеної відповіді, надаючи гравцеві зворотний зв'язок та, у разі успіху, переводить його на наступний етап.

Переваги:

- має власний калькулятор;
- стильний дизайн оформлення гри.

Недоліки:

- не зрозумілий, як для початківця інтерфейс;
- доволі складні для користувача математичні рівності;
- занадто багато реклами, як для гри;
- немає веб-версії.

Math Fight – це динамічна математична гра, розроблена для змагання двох гравців у реальному часі. Її основне призначення – розвиток швидкості мислення, уважності та математичних навичок у гравців різного віку. Гра побудована за принципом швидкої реакції на запропоновані приклади, де перемогу здобуває той, хто правильно відповість на більшу кількість завдань за відведений час [4].

Принцип роботи полягає в генерації випадкових математичних прикладів відповідно до заданих параметрів складності. Кожна правильна відповідь миттєво враховується в особистому рахунку гравця. Для контролю відповідності результатів використовується автоматична перевірка введених значень та система підрахунку балів у реальному часі (рис. 2.).

Переваги:

- зручний інтерфейс;
- вибір режиму гри;
- інтуїтивно зрозуміла гра.

Недоліки:

- немає веб-версії;
- гра розрахована на двох гравців;
- багато реклами, як для гри.

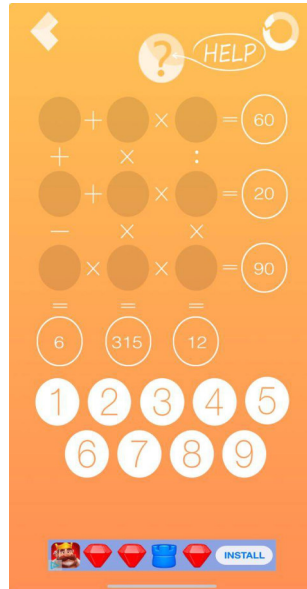


Рис. 1. Інтерфейс гри Math Puzzel

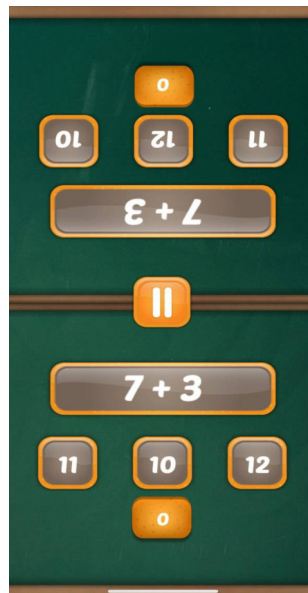


Рис. 2 Інтерфейс гри Math Fight

Mathemagics – це освітньо-розважальна гра, створена для тренування швидкості математичних обчислень та розвитку пам'яті [5]. Вона орієнтована на широке коло користувачів – від дітей, які навчаються основам арифметики, до дорослих, що прагнуть підтримувати інтелектуальну активність. Гра поєднує елементи гейміфікації з класичними математичними вправами, що робить навчання більш цікавим та мотивуючим.

Mathemagics має адаптивну платформу, що дозволяє запускати гру на різних пристроях – від смартфонів та планшетів (рис. 3.). Завдяки цьому вона може бути використана як у навчальних закладах, так і для індивідуальних тренувань. Гра поєднує

навчальний та розважальний елементи, сприяючи розвитку уваги, логічного мислення та швидкості обчислень.

Переваги:

- інтерфейс розроблений вигляді мультфільму;
- багато режимів гри;
- цікаве подання рівностей.

Недоліки:

- сама гра платна, є безкоштовна версія, але її не достатньо для гри;
- хоч і інтерфейс вигляді мультфільму, але для початкового користувача
- складно.
- немає веб-версії.

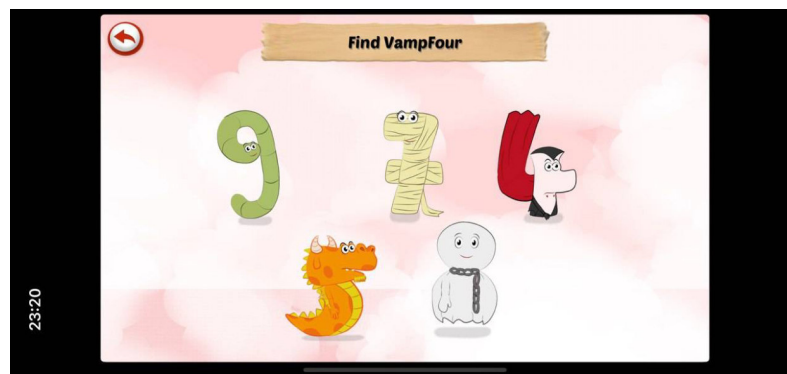


Рис 3. Інтерфейс гри Mathemagics

4. Результат аналізу ігрових рішень для покращення когнітивної стимуляції

Розглянувши ігри представлені вище, встановлені такі характерні ознаки:

- більшість ігор має власний парсер для обчислення алгебраїчних рівностей;
- більшість подібних ігор мають адаптацію під мобільні пристрої;

Також виявились недоліки подібних ігор, наприклад:

- всі наведені ігри не мають веб-версії;
- наведені безкоштовні ігри мають важкий для початкового користувача інтерфейс і за багато реклами;
- платні ігри, мають складний для початкового користувача інтерфейс.

5. Визначення структури веб-застосунку

На основі оцінки вищезазначених систем, було прийняте рішення про створення власної гри для когнітивної стимуляції у вигляді веб-застосунку. Веб-застосунок реалізований в ході роботи має такі властивості:

Технічні вимоги:

- веб-застосунок з кросплатформенною підтримкою;
- адаптивний дизайн для мобільних пристроїв;
- власний ефективний парсер для алгебраїчних рівностей;
- архітектура REST API.

Функціональні вимоги:

- система рівнів з поступовим зростанням складності;
- автоматична генерація різноманітних завдань;
- детальна аналітика прогресу користувача;
- можливості індивідуального тренування.

Вимоги до користувацького досвіду:

- простий та інтуїтивний інтерфейс;
- відсутність реклами, що заважає концентрації;
- швидкий час відгуку системи;
- зручність використання на різних пристроях.

6. Архітектура веб-застосунку

Для досягнення результату в ході роботи, використано наступні технології:

Backend розробка:

- Python 3.10 – основна мова програмування [6];
- C – мова загального призначення (в роботі використовується для створення static data) [7];
- MyPyc – бібліотека для компілювання Python модуль в C розширення [8];
- Flask Framework - контейнер на основі якого розгортається сервер [9];
- REST API – визначення загальної архітектури сервера [10].

Frontend розробка:

- JavaScript – мова програмування, яка найчастіше використовується для створення інтерактивних Web-сторінок [11];
- HTML5 – мова розмітки гіпертексту [12];
- CSS3 – спеціальна мова, що використовується для запису оформлення сторінок, написаних мовами розмітки даних [13].

Допоміжні технології:

- JSON – обмін інформацією з сервером [14];
- WebStorm – середовище розробки з підтримкою Python, JavaScript, HTML, CSS, C [15].

7. Архітектура REST API

Application Programming Interface (API) – це набір визначень і протоколів для створення та інтеграції програмного забезпечення (рис. 4). Цей архітектурний підхід обертається навколо надання програмного інтерфейсу набору послуг для різних програм, які обслуговують різних споживачів.

При використанні архітектурного підходу в контексті веб-розробки, API визначається як набір специфікацій, таких як передача повідомлення запиту до ресурсу, за допомогою Hypertext Transfer Protocol (HTTP), визначенням структури запитів-відповідей, з використанням технологій Extensible Markup Language (XML) або JavaScript Object Notation (JSON).

Representational State Transfer (REST) – це набір архітектурних обмежень, а не протокол чи стандарт. Розробники API можуть реалізувати REST різними способами.

Коли відбувається запит до сервера чи програмного забезпечення за допомогою REST API, клієнт передає інформацію про стан ресурсу, та подає інформацію у одному з форматів через HTTP: JSON, XML, JavaScript-запит, звичайний текст, тощо.

Важливо розуміти дещо про використання HTTP у REST API: заголовки та параметри відіграють важливу роль у обміні інформацією між ресурсами, оскільки вони містять важливу інформацію ідентифікатора про метадані запиту, авторизацію, тип контенту, єдиний ідентифікатор ресурсу, кешування, куки, код відповідей, інше. Існують заголовки запитів та відповідей, кожен з них містять власну інформацію про з'єднання HTTP та коди станів.

Найбільш популярний варіант надсилання тіла інформації – це JSON, оскільки він не прив'язаний до певної мови програмування, зручний у використанні, легкий у

розумінні як людьми, так і машинами. У своїй основі містить пари — ключ та значення, які використовуються у формуванні тіл запитів та відповідей, у вигляді масивів відповідних пар (рис. 4.).

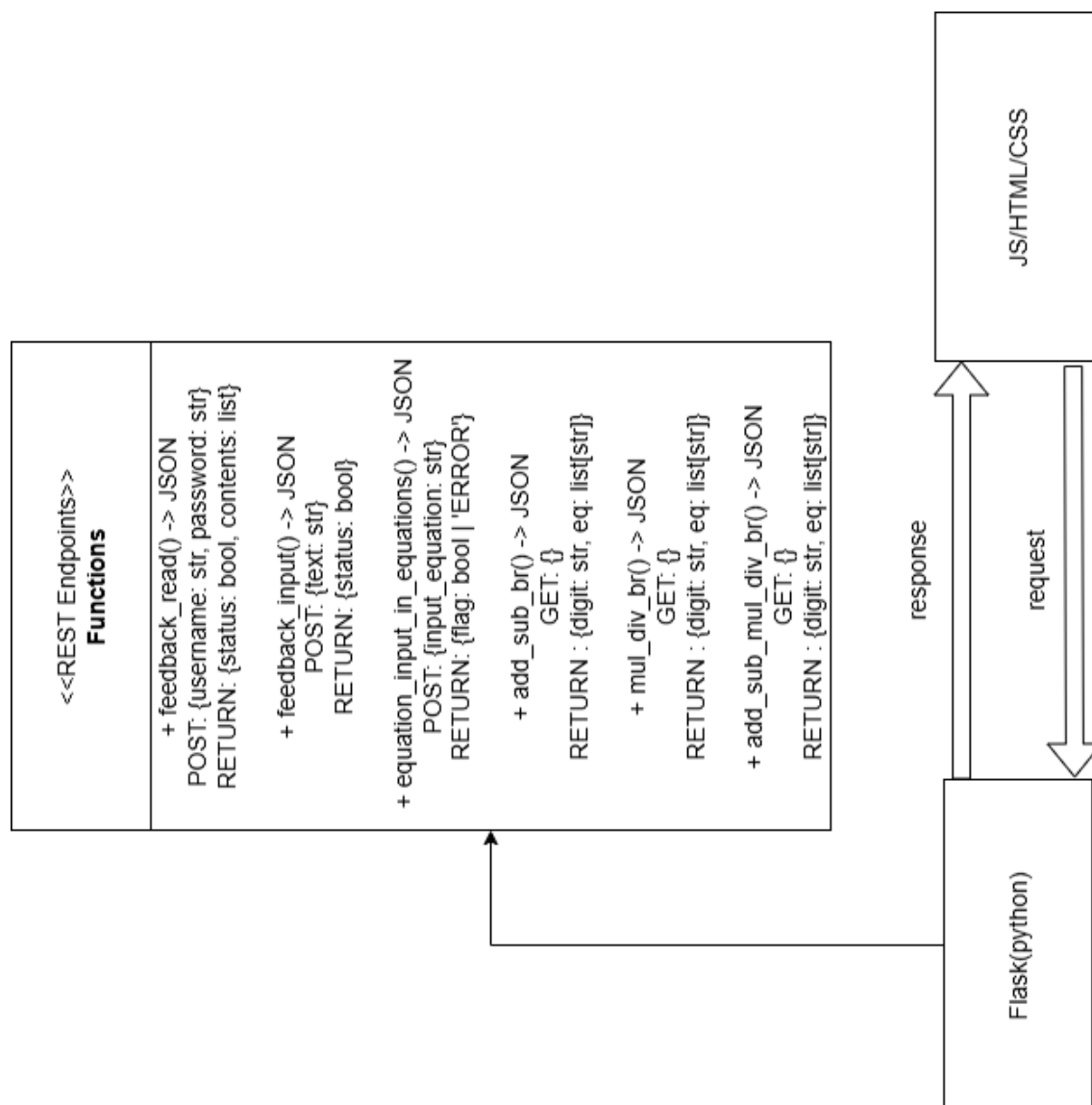


Рис. 4. UML діаграма серверної частини REST API

Для того, щоб API розглядався як REST, має слідувати таким критеріям: архітектура, що складається з клієнтів, серверів та ресурсів, із запитам, які використовують протокол HTTP; всі комунікації між клієнтом та сервером відбуваються без зберігання стану та сесій, кожен запит та відповідь є відокремленими та незалежним; дані кешуються, для полегшення навантаження на сервер; стандартизація запитів та відповідей між клієнтом та сервером відповідно; багатшарова система з розподіленням обов'язків, ієрархічна структура невидима для клієнтів.

8. Ролі в комп'ютерній системі

Сервер:

- генерація випадкових чисел та відповідних рівностей;
- перевірка правильності введених користувачем рівностей;

- збереження статистики та прогресу;
- обробка зворотного зв'язку.
Користувач:
 - вибір рівня складності гри;
 - введення алгебраїчних рівностей;
 - отримання результатів та статистики;
 - надання зворотного зв'язку.Розробник:
 - моніторинг роботи системи;
 - аналіз відгуків користувачів;
 - оновлення та вдосконалення системи.

Для того, щоб перевірити чи рівність правильна потрібно спочатку пропустити вхідні символи через лексичний аналізатор, який створить токени потім на базі математичних операції побудувати дерево і рекурсивно його обходити з поступовим зменшенням пріоритету математичних операції. Дерево обходиться згідно правилами пріоритету операції ('()', '^', '*', '/', '+', '-'). Рекурсивна оцінка вузлів дерева дозволяє отримати значення виразу та перевірити, чи рівність є коректною (ліва частина == права частина) [16].

Рекурсія – це процес визначення чогось у межах самого себе. Прикладному світі може бути розміщення двох дзеркал одне напроти одного. Будь-який об'єкт між ними відображатиметься рекурсивно [17].

Токен – об'єкт, що утворюється із лексеми в процесі лексичного аналізу.

Лексер, або лексичний аналізатор – це комп'ютерна програма або частина програми, яка перетворює вхідний потік символів (в нашому випадку математична рівність) на послідовність лексем або токенів [18].

Абстрактне синтаксичне дерево (AST) – це дерево представлення вихідного коду комп'ютерної програми, яке передає структуру вихідного коду. Кожен вузол у дереві представляє конструкцію, що зустрічається у вихідному коді [19].

Парсер – це програмний компонент, який приймає вхідні дані (часто текст) і створює структуру даних – часто дерево розбору, абстрактне дерево синтаксису в процесі.

9. Основний алгоритм

Основний алгоритм, який буде генерувати всі можливі математичні рівності з випадкового числа, побудовано на основі власного алгоритму (рис. 5, 6). В основі розробки покладено парсер.

Алгоритм генерування алгебраїчних рівностей:

Крок 1. Ініціалізація

- 1.1 Задаємо максимальну кількість ітерації для перестановок цифр в числі і комбінацій математичних операторів (в нашому випадку задаємо таке число, яке буде більше за $4! * 4^4$ але менше за $8! * 8^4$).
- 1.2 Задаємо кількість чисел, для кожного рівня.
- 1.3 Задаємо мінімальну кількість рівностей для кожного числа.
- 1.4 Задаємо шаблони для створення рівностей вигляді генераторів.

Крок 2. Створюємо ітерабельний об'єкта - всі можливі перестановки цифр в числі.

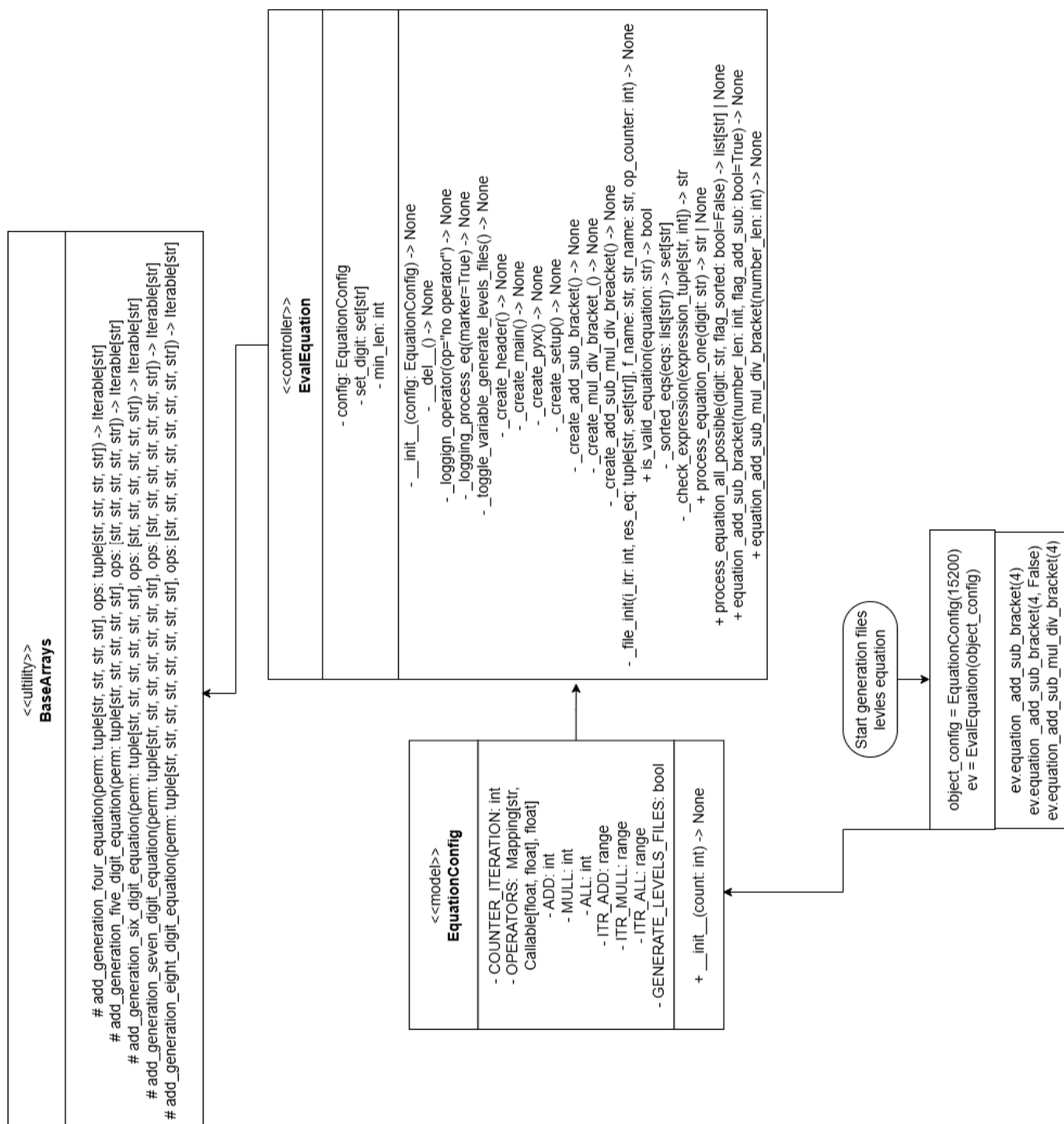


Рис. 5. UML діаграма структури коду для генерува алгебраїчних рівностей із застосуванням архітектурного підходу SOLID.

- Крок 3. Створюємо ітерабельний об’єкта - всі можливі комбінації математичних операторів.
- Крок 4. Передаємо в шаблони пари цифр і пари комбінацій математичних операторів, отримуємо математичні рівності.
- Крок 5. Прибираємо дублікати.
- Крок 6. Для кожної рівності в ітерації і-тий елемент замінюємо одним математичним оператором на знак дорівнює.

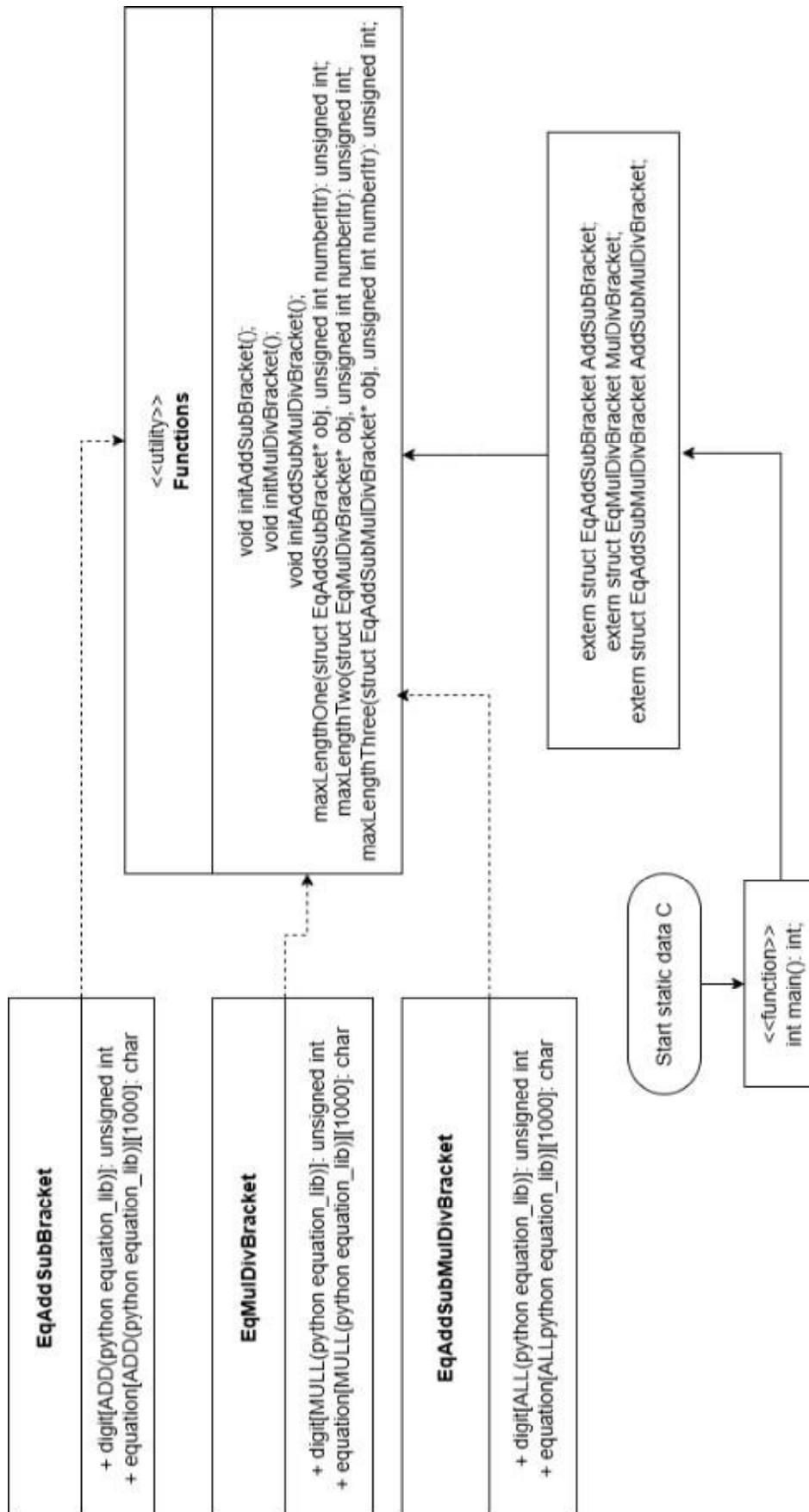


Рис. 6. UML діаграма структури коду для перевірки максимальної довжини рівності і відладки структури static data

Крок 7. Перевіряємо всі математичні рівності через парсер з рекурсивним обходом дерева з поступовим пониженням математичного пріоритету операції.

Крок 8. Після знаходження всіх алгебраїчних рівностей створюємо заголовковий файл h.

Крок 9. Створюємо файл для перевірки максимальної довжини математичної рівності.

Крок 10. Створюємо проміжний файл між пайтоном і C.

Крок 11. Створюємо файл setup.

Крок 12. Збираємо головний файл C для відладки заголовочного файлу h та рух.

Крок 13. Змінюємо довжину масиву, як в заголовочному так і в проміжному файлі.

Крок 14. Збираємо все під пайтон модуль.

10. Аналіз використання оперативної пам'яті при генерації алгебраїчних рівностей

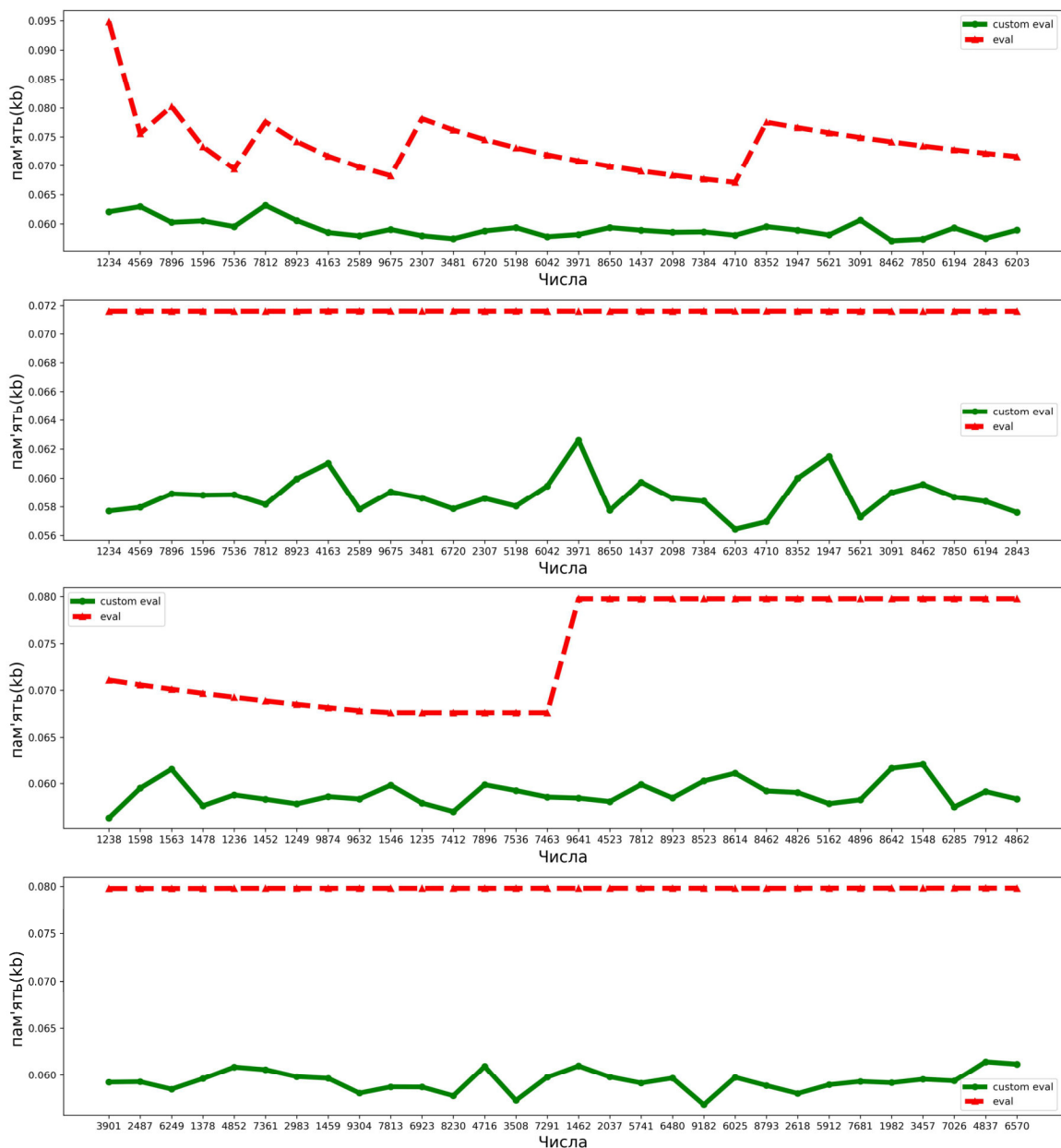


Рис. 7. Порівняння використання оперативної пам'яті CUSTOM_EVAL і EVAL

Аналіз проводився за допомогою бібліотеки Tracemalloc (python). Для порівняння використано тільки вбудовану функцію eval, порівнювати з іншими бібліотеками немає сенсу оскільки вони використовують більше оперативної пам'яті ніж власна реалізація при перевірці однієї рівності. При перевірці вже багатьох алгебраїчних рівностей вони переповнюють оперативну пам'ять. Слід зазначити, що аналіз проведено декілька разів з різними числами для генерації рівностей (рис. 7.).

Проведений аналіз використання оперативної пам'яті показує, що вбудована функція eval в Python за рахунок постійного компілювання в байткод не може перевіряти масові рівності при їх створенні оскільки іде переповнення оперативної пам'яті при перевірці. Коли кількість рівностей зростає Cpython починає робити алокації за рахунок яких іде надмірне споживання оперативної пам'яті у вигляді стрибків. Власна бібліотека немає вказаних вище мінусів, і використовує більше стабільніший профіль оперативної пам'яті (рис. 8.).



Рис. 8. Результат порівняння аналізу використання оперативної пам'яті при генерації алгебраїчних рівностей CUSTOM_EVAL і EVAL

Висновок

У статті розглянуто відомі ігри для тренування когнітивної стимуляції, а саме: Math Puzzle, Math Fight, Mathemagics. Представлено опис переваг та недоліків кожної системи.

Розроблений власний веб-застосунок з адаптацією під мобільні пристрої. Веб-застосунок використовує власний парсер з рекурсивним обходом абстрактного дерева, з поступовим пониженням математичного пріоритету. Власна бібліотека була протестована на використання оперативної пам'яті з вбудованою функцією eval в Python.

У результаті аналізу показано, що вбудована функція робить багато алокації (або виділення) в оперативної пам'яті, що видно з вище наведених графіків.

Практичне значення роботи полягає у можливості застосування створеного інструменту для підтримки та розвитку когнітивних функцій як у людей похилого віку, так і серед молоді. Також робота має теоретичне значення, яке визначається формалізацією методу побудови рівностей із використанням AST та аналізом ефективності алгоритмів у контексті споживання ресурсів.

Подяки. Автори висловлюють подяку А.М. Гусаку за допомогу у постановці задачі.

Список використаної літератури:

1. The Impact of Action Video Game Training on Mathematical Abilities in Adults. [Електронний ресурс] – Режим доступу: <https://journals.sagepub.com/doi/full/10.1177/2332858417740857>.
2. The effects of two digital educational games on cognitive and non-cognitive math and reading outcomes. [Електронний ресурс] – Режим доступу: <https://www.sciencedirect.com/science/article/abs/pii/S0360131519302337>.
3. Influence of game-based learning in mathematics education on the students' cognitive and affective domain: A systematic review. [Електронний ресурс] - Режим доступу: <https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2023.1105806/full>
4. Game-Based Training to Promote Arithmetic Fluency. [Електронний ресурс] – Режим доступу: <https://www.frontiersin.org/journals/education/articles/10.3389/feduc.2019.00118/full>.
5. Evaluating the effectiveness of a game-based rational number training - In-game metrics as learning indicators. [Електронний ресурс] – Режим доступу: <https://www.sciencedirect.com/science/article/pii/S0360131518300125>.
6. Python 3.10. [Електронний ресурс] – Режим доступу: <https://www.python.org/downloads/release/python-3100/>.
7. C. [Електронний ресурс] – Режим доступу: <https://www.c-language.org/>.
8. Мурс. [Електронний ресурс] – Режим доступу: <https://mypyc.readthedocs.io/en/latest/introduction.html>.
9. Flask. [Електронний ресурс] – Режим доступу: <https://flask.palletsprojects.com/en/stable/>.
10. What is REST API? [Електронний ресурс] – Режим доступу: <https://www.redhat.com/en/topics/api/what-is-a-rest-api>.
11. JavaScript. [Електронний ресурс] – Режим доступу: <https://css.in.ua/js/objects>.
12. HTML5. [Електронний ресурс] – Режим доступу: https://css.in.ua/article/shcho-take-css_3.
13. CSS3. [Електронний ресурс] – Режим доступу: <https://css.in.ua/css/properties>.
14. JSON. [Електронний ресурс] – Режим доступу: <https://www.json.org/json-en.html>.
15. WebStrom. [Електронний ресурс] – Режим доступу: <https://www.jetbrains.com/ru-ru/webstorm/>.
16. Math-Equation-lib. [Електронний ресурс] – Режим доступу: <https://github.com/Sasha376slysenko/Math-Equation-lib>.
17. Рекурсія в Python. [Електронний ресурс] – Режим доступу: <https://acode.com.ua/recursion-python/>.
18. Лексичний аналізатор. [Електронний ресурс] – Режим доступу: <https://www.guru99.com/uk/compiler-design-lexical-analysis.html>.
19. Deepsource // AST. [Електронний ресурс] – Режим доступу: <https://deepsources.com/blog/python-asts-by-building-your-own-linter>.

References:

1. The Impact of Action Video Game Training on Mathematical Abilities in Adults. [Electronic resource] – Access mode: <https://journals.sagepub.com/doi/full/10.1177/2332858417740857>.
2. The effects of two digital educational games on cognitive and non-cognitive math and reading outcomes. [Electronic resource] – Access mode: <https://www.sciencedirect.com/science/article/abs/pii/S0360131519302337>.
3. Influence of game-based learning in mathematics education on the students' cognitive and affective domain: A systematic review. [Electronic resource] – Access mode: <https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2023.1105806/full>
4. Game-Based Training to Promote Arithmetic Fluency. [Electronic resource] – Access mode: <https://www.frontiersin.org/journals/education/articles/10.3389/feduc.2019.00118/full>.
5. Evaluating the effectiveness of a game-based rational number training - In-game metrics as learning indicators. [Electronic resource] – Access mode: <https://www.sciencedirect.com/science/article/pii/S0360131518300125>.
6. Python 3.10. [Electronic resource] – Access mode: <https://www.python.org/downloads/release/python-3100/>.
7. C. [Electronic resource] – Access mode: <https://www.c-language.org/>.
8. Мурс. [Electronic resource] – Access mode: <https://mypyc.readthedocs.io/en/latest/introduction.html>.
9. Flask. [Electronic resource] – Access mode: <https://flask.palletsprojects.com/en/stable/>.
10. What is REST API? [Electronic resource] – Access mode: <https://www.redhat.com/en/topics/api/what-is-a-rest-api>.
11. JavaScript. [Electronic resource] – Access mode: <https://css.in.ua/js/objects>.
12. HTML5. [Electronic resource] – Access mode: https://css.in.ua/article/shcho-take-css_3.
13. CSS3. [Electronic resource] – Access mode: <https://css.in.ua/css/properties>.
14. JSON. [Electronic resource] – Access mode: <https://www.json.org/json-en.html>.
15. WebStrom. [Electronic resource] – Access mode: <https://www.jetbrains.com/ru-ru/webstorm/>.

16. Math-Equation-lib. [Electronic resource] – Access mode: <https://github.com/Sasha376slysenko/Math-Equation-lib>.
17. Recursion in Python.. [in Ukrainian] [Electronic resource] – Access mode: <https://acode.com.ua/recursion-python/>.
18. Lexical analyzer. [in Ukrainian] [Electronic resource] – Access mode: <https://www.guru99.com/uk/compiler-design-lexical-analysis.html>.
19. Deepsource // AST. [Electronic resource] – Access mode: <https://deepsources.com/blog/python-asts-by-building-your-own-linter>.

LYSENKO Oleksandr Volodymyrovych,

Student majoring in Applied Physics and Nanomaterials at Bohdan Khmelnytsky National University of Cherkasy, Ukraine

TATARCHUK Yevgeny Viktorovich,

Candidate of Physical and Mathematical Sciences, Associate Professor of the Department of Physics, Bohdan Khmelnytsky National University of Cherkasy, Ukraine

ALGORITHM AND WEB APPLICATION FOR AUTOMATIC GENERATION OF ALGEBRAIC EQUATIONS FOR COGNITIVE TRAINING

Summary. Introduction. *In modern society, maintaining cognitive functions has become increasingly important. Older people face gradual memory decline and reduced thinking speed, while younger generations tend to lose mental arithmetic skills due to excessive reliance on calculators and digital technologies. Existing games for cognitive stimulation (Math Puzzle, Math Fight, Mathemagics) demonstrate the potential of this approach, but share common limitations: no web versions, overloaded interfaces, or strict paid-use conditions, which reduce their accessibility for a wide range of users.*

Purpose. *The aim of this work is to analyze existing games for cognitive stimulation, to develop a proprietary cross-platform web application, and to describe a custom library for automatic generation and verification of algebraic equations based on an Abstract Syntax Tree (AST) parser.*

Results. *Three well-known cognitive training games were analyzed and their advantages and disadvantages identified. Based on this analysis, a web application was developed with cross-platform and mobile-adaptive support, a REST API architecture, a level system with gradually increasing difficulty, and an ad-free interface. The core of the application is a proprietary algorithm for generating all valid algebraic equations from a given number, implemented using a recursive AST parser with progressive reduction of operator precedence. To optimize performance, the library was compiled from Python to a C extension using MyPyc. A comparative RAM consumption analysis using the Tracemalloc library demonstrated that the built-in Python eval function produces excessive memory allocations when processing large sets of equations, whereas the custom library maintains a significantly more stable memory profile.*

Conclusion. *A web application for cognitive training was developed with adaptation for mobile devices. A custom algorithm and library for generating and verifying algebraic equations were implemented and tested. The proprietary AST-based parser outperforms the built-in eval function in memory efficiency when handling large equation arrays. The practical value of the work lies in its applicability for supporting and developing cognitive functions in both elderly people and youth. The theoretical contribution consists in the formalization of an equation-generation method using AST and the analysis of algorithm efficiency in terms of resource consumption.*

Keywords: *cognitive stimulation, web application, token, recursion, AST, API.*

*Одержано редакцією 21.07.2025 р.
Прийнято до публікації 24.09.2025 р.*