

Кафедра інформатики

## **КУРСОВА РОБОТА**

**Тема: “ ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕРАКТИВНИХ ФОРМУЛ”**

**Виконала:**

Хейфець Юнна Едуардівна,  
учениця 10-А класу

**Науковий керівник:**

Хижа Олександра Леонідович,  
Канд. фіз.-мат. наук, доцент ДНУ

**Робота допущена до захисту**

\_\_\_\_\_ ПІБ керівника

## ЗМІСТ

Вступ.....	4
1. Теоретичні основи роботи.....	5
1.1. Означення формальних мов і граматик.....	5
1.2. Класифікація граматик.....	6
1.3. Нотація Бекуса-Наура.....	7
1.4. Синтаксичний аналіз .....	8
1.5. Рекурсивний спуск.....	10
2. Існуючі інструменти для роботи з формулами .....	11
3. Розробка вебзастосунку.....	13
3.1. Загальний опис програми та її реалізації .....	13
3.2. Опис формули .....	14
3.2.1. Формальна граматика.....	14
3.2.2. Структура даних .....	15
3.2.3. Перетворення структури даних.....	17
3.3. Дії з формулами .....	17
3.3.1. Перетворення рівняння.....	18
3.3.2. Перетворення виразів.....	19
3.4. Інструкція для користувача .....	20
3.4.1. Елементи управління програмою .....	21
3.4.2. Додавання елементів до інтерактивного поля.....	21
3.4.3. Дії з формулами .....	23
3.4.4. Режим додавання нової частини .....	25

3.4.5. Інші дії програми .....	26
4. Застосування результатів розробки.....	28
4.1. Застосування в математиці .....	28
4.2. Застосування в фізиці .....	29
Висновок .....	32
Список літератури.....	33
Додатки .....	34
Додаток А .....	34
Додаток Б .....	35

## Вступ

У сучасному світі технології швидко розвиваються, що дозволяє нам змінювати наше сприйняття та спосіб роботи з інформацією. У сфері освіти, технології також стають все більш важливим і необхідним компонентом навчального процесу. Вирішення задач на папері зазвичай було одним із засобів навчання та перевірки знань. Однак, з поширенням дистанційного навчання та зростанням кількості школярів, які навчаються онлайн, виконання таких завдань на папері може бути не зручним.

Тому, у цій курсовій роботі, розробляється застосунок, який допомагає школярам та вчителям вирішувати будь-які задачі, в яких використовуються формули. Цей застосунок дозволяє виконувати перетворення, аналогічні тим, що ми робимо на папері, але в електронному вигляді у зручний спосіб.

Метою даної курсової роботи є розробка застосунку, що допоможе школярам та вчителям вирішувати задачі пов'язані з математичними формулами. Основним завданням дослідження є аналіз існуючих методів перетворення математичних формул, а також розробка та реалізація алгоритмів, що дозволяють виконувати перетворення. Результатом дослідження буде функціональний застосунок, який забезпечить школярів та вчителів зручним та ефективним інструментом для виконання математичних перетворень.

Відповідно до мети роботи були поставлені наступні завдання:

1. Вивчити теоретичні основи формальних мов і граматик
2. Ознайомитися з існуючими інструментами для роботи з формулами
3. Розробити вебзастосунок
4. Розглянути застосування результатів розробки

## 1. ТЕОРЕТИЧНІ ОСНОВИ РОБОТИ

### 1.1. Означення формальних мов і граматик

У формальних мовних **алфавітах** називають непорожню множину символів. **Словом** в алфавіті може бути будь-яка скінченна послідовність символів. Порожнім словом  $\epsilon$  називають послідовність символів нульової довжини.

Якщо заданий алфавіт  $\Sigma$ , то множину усіх слів в алфавіті  $\Sigma$  позначають  $\Sigma^*$ . Наприклад, якщо  $\Sigma = \{a, b\}$ , то  $\Sigma^* = \{ \epsilon, a, b, aa, ab, bb, ba, aaa \dots \}$ .

**Мовою** в алфавіті  $\Sigma$  називають множину слів в алфавіті, тобто мов є підмножиною  $\Sigma^*$ .

**Визначення.** Формальною граматикою  $G$ , що породжує множину символів, називається наступна сукупність чотирьох об'єктів:  $G = \{ \Sigma, N, P, S \}$ , де

- $\Sigma$  – алфавіт термінальних символів; терміналом може бути будь-який об'єкт, який не змінює своє значення; зазвичай в якості терміналів використовуються ASCII або Unicode символи.
- $N$  – алфавіт нетермінальних символів (іноді їх називають *синтаксичними змінними*); з нетерміналів будуються ланцюжки, породжувані граматикою.
- $P$  – множина правил або продукцій вигляду  $\alpha \rightarrow \beta$ , де  $\alpha$  і  $\beta$  – слова з алфавіту  $\Sigma \cup N$ .
- $S$  – початковий символ граматичної;  $S \in N$ .

Ми можемо сприймати продукції в граматичній, як способи «перетворення» одних слів з алфавіту  $\Sigma \cup N$  в інші слова з  $\Sigma \cup N$ .

**Визначення.** Ми можемо *безпосередньо вивести*  $v$  з  $u$  в граматичній  $G = \{ \Sigma, N, P, S \}$  (позначається  $u \Rightarrow v$ ), якщо:

- $y \rightarrow y' \in P$
- $x, z \in (\Sigma \cup N)^*$
- $u = xyz$  та  $v = xy'z$ .

Ми можемо *вивести*  $v$  з  $u$  (позначається  $u \Rightarrow^* v$ ) в граматиці  $G = \{\Sigma, N, P, S\}$ , якщо  $u \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \dots w_n \Rightarrow v$ . Таке перетворення позначається  $u \Rightarrow^* v$ .

## 1.2. Класифікація граматик

За ієрархією Чомські граматики поділяються на 4 типи. Кожен наступний тип є підтипом попереднього. Далі будемо використовувати малі літери  $a, b, c, \dots$  для позначення терміналів, великі літери  $A, B, C, \dots$  – для нетерміналів, грецькі літери  $\alpha, \beta, \gamma, \dots$  – для послідовностей з терміналів та нетерміналів.

**Тип 0.** Необмежені. На продукції не накладаються ніякі обмеження. Продукції мають вигляд  $\alpha \rightarrow \beta$ .

**Тип 1.** Контекстно-залежні. Ліва частина може містити нетермінал, оточений послідовністю символів, які присутні в правій частині. Сам термінал замінюється послідовністю символів. Продукції мають вигляд  $\alpha A \beta \rightarrow \alpha \gamma \beta$ .

**Тип 2.** Контекстно-вільні. Ліва частина складається з одного нетерміналу. Продукції мають вигляд  $A \rightarrow \beta$ .

**Тип 3.** Регулярні граматики. Продукції мають вигляд  $A \rightarrow \alpha A$  або  $A \rightarrow \alpha$

**Приклад 1.** Наступна контекстно-вільна граматика генерує правильні вирази, які складаються зі знаків додавання та множення, групуючих дужок та змінних  $x$  та  $y$ .

Термінальний алфавіт  $\Sigma = \{x, +, *, (, )\}$

Нетермінальний алфавіт –  $N = \{E, T, F, V\}$

Продукції  $P = \{$

$$E \rightarrow T$$

$$E \rightarrow T + E$$

$$T \rightarrow F$$

$$T \rightarrow F * T$$

$$F \rightarrow (E)$$

$$F \rightarrow V$$

$$V \rightarrow x$$

$$V \rightarrow y$$

$\}$

Стартовий нетермінал –  $E$

Виведемо вираз  $y^*(x+y)$  за допомогою перерахованих правил.

$$E \Rightarrow T \Rightarrow F * T \Rightarrow V * T \Rightarrow y^* T \Rightarrow y^* F \Rightarrow y^* (E) \Rightarrow y^* (T + E) \Rightarrow y^* (F + E) \Rightarrow y^* (V + E) \\ \Rightarrow y^* (x + E) \Rightarrow y^* (x + T) \Rightarrow y^* (x + F) \Rightarrow y^* (x + V) \Rightarrow y^* (x + y)$$

### 1.3. Нотація Бекуса-Наура

При описі складних конструкцій за допомогою формальних граматик доведеться використовувати багато нетермінальних символів, через це такий опис втрачає наочність. В цьому можна використати форму Бекуса-Наура (БНФ). В БНФ в якості нетерміналів використовуються слова природної мови, узяті в кутові дужки, а в якості знака переходу від лівої частини продукції до правої – дві двокрапки та знак дорівнює. Наприклад, продукцію

$$E \rightarrow N + N$$

де  $E$  позначає «вираз», а  $N$  – число, можна записати наступним чином:

$$\langle \text{додавання} \rangle ::= \langle \text{число} \rangle + \langle \text{число} \rangle$$

Якщо декілька продукцій мають однакові ліву частину, то їх можна об'єднати в одну продукцію, в правій частині якої – праві частини об'єднаних продукцій, розділені знаком «|». Наприклад, продукції

$\langle \text{змінна} \rangle ::= a$

$\langle \text{змінна} \rangle ::= b$

можна поєднати в

$\langle \text{змінна} \rangle ::= a \mid b$

Зазвичай за допомогою БНФ описують контекстно-вільні граматики, але також її можна використовувати для регулярних граматик.

#### 1.4. Синтаксичний аналіз

Задачею синтаксичного аналізу є зіставлення формальної граматики з послідовністю символів в природній або формальній мові. В процесі синтаксичного аналізу створюється дерево розбору або структура даних, яка представляє синтаксичну структуру вхідної послідовності, і яку зручно потім обробляти.

Дерево синтаксичного розбору – це впорядковане кореневе дерево, яке зображує синтаксичну структуру рядка відповідно до формальної граматики. В такому дереві листя зображують нетермінали, а внутрішні вузли – термінали.



**Приклад 2.** На *Помилка! Джерело посилання не знайдено.* можна побачити приклад синтаксичного розбору виразу  $x + y * (x + y)$  в дерево синтаксичного розбору відповідно до граматики, описаної в Приклад 1.

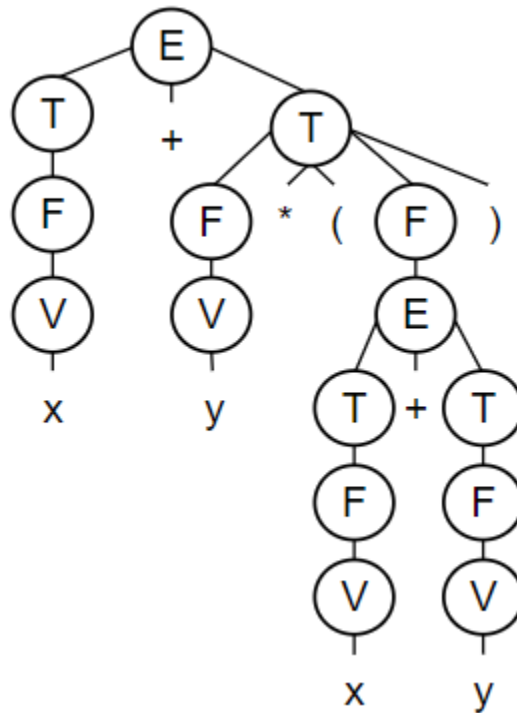


Рисунок 1 Дерево синтаксичного розбору виразу  $x + y * (x + y)$

Алгоритми синтаксичного аналізу поділяють на такі групи:

- Спадні (низхідні). Починаючи зі стартового нетерміналу намагаємось перетворити усі нетермінали, використовуючи продукції зліва-направо, поки не отримаємо потрібну послідовність терміналів. Іншими словами, будуємо дерево починаючи з коріння та закінчуючи листями.
- Висхідні. Для кожного терміналу використовуємо продукції справа-наліво, потім виконуємо такі ж дії для отриманих нетерміналів, поки не залишиться стартовий нетермінал. Іншими словами, будуємо дерево починаючи з листів та закінчуючи корінням.

### 1.5. Рекурсивний спуск

Програма синтаксичного аналізу методом рекурсивного спуску складається із набору процедур: по одній процедурі для кожного нетерміналу. Робота програми починається із виклику процедури для початкового символу і успішно завершується у випадку сканування всієї вхідної стрічки. Нетермінали розкриваються послідовно зліва-направо.

Варто зауважити, що рекурсивний спуск є вразливим для граматик, в яких в лівій частині продукції та напочатку правої частини стоїть один і той же нетермінал. В такому випадку програма-аналізатор зациклюється. Це відбувається через те, що процедура, намагаючись розкрити поточний нетермінал, викликає таку ж процедуру, яка намагається розкрити той же нетермінал.

При аналізі методом рекурсивного спуску може виникнути проблема, яка полягає в тому, що неможливо визначити, за яким правилом перетворювати поточний нетермінал. В такому випадку можна перебирати усі правила, в яких в правій частині стоїть поточний нетермінал, та визначати чи відповідають нетермінали в тілі продукції вхідному рядку. Якщо не відповідають, то потрібно повернутись та обрати іншу продукцію.

Щоб уникнути проблем, пов'язаних з «поверненнями», використовують LL(k) граматиками. Такі граматиками дозволяють, однозначно визначити необхідну продукцію, переглянувши лише k поточних символів.

## 2. ІСНУЮЧІ ІНСТРУМЕНТИ ДЛЯ РОБОТИ З ФОРМУЛАМИ

Існує кілька доступних програмних інструментів, які дозволяють користувачам вирішувати математичні задачі та працювати з формулами або рівняннями.

Розглянемо найбільш відомі програмні продукти, які допомагають ефективно й точно виконувати складні різноманітні математичні завдання:

- **Maple** — це система комп'ютерної алгебри, яка дозволяє користувачам виконувати символічні та числові обчислення. Maple використовується в різних сферах, включаючи математику, фізику, фінанси та освіту. Програма має велику бібліотеку функцій і

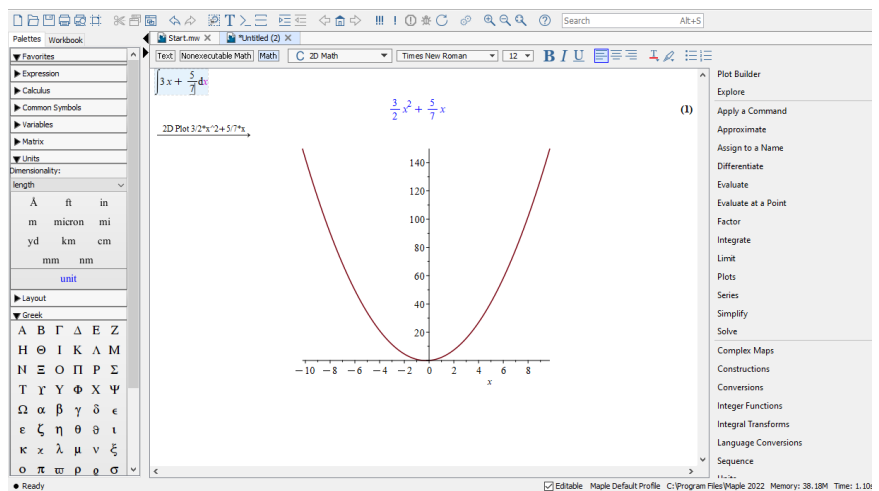


Рисунок 2. Інтерфейс Maple

пакетів, які дозволяють легко виконувати обчислення без необхідності писати код. Взаємодіяти з програмою можна за допомогою командного рядка або зручного графічного інтерфейсу. Однак інтерфейс може бути складним у використанні для тих, хто не знайомий з математичними програмними.

- **Mathematica** також є програмним пакетом, який використовується для математичних обчислень, аналізу даних і візуалізації. Mathematica має інтерфейс записної книжки, та дозволяє користувачам створювати та впорядковувати свої обчислення. Обчислення виконуються за допомогою введення команд. До кожної команди надається документація з прикладами. Однак деяким користувачам

такий інтерфейс може здатися занадто складним через кількість доступних функцій і опцій.

- **MATLAB** — це мова програмування, а також середовище для числових обчислень. **MATLAB** дозволяє маніпулювати матрицями, будувати графіки, виконувати симуляції, створювати інтерфейси користувача, а також взаємодіяти з програмами, написаними іншими мовами. Використовувати **MATLAB** можна за допомогою

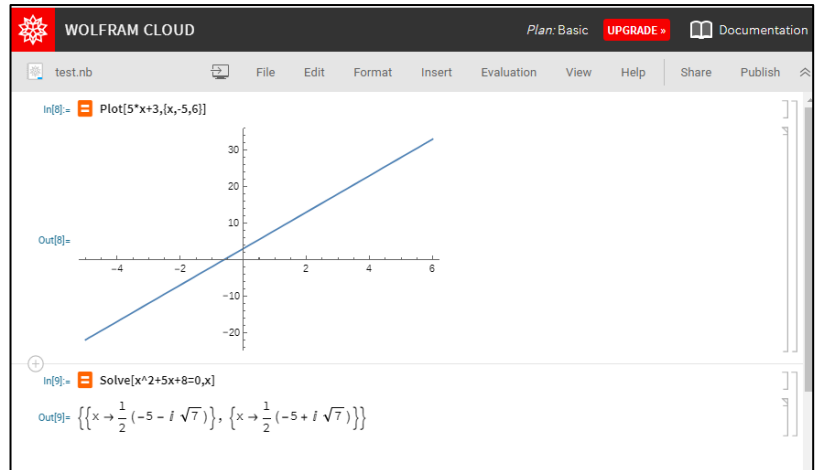


Рисунок 3 Вебінтерфейс Mathematica

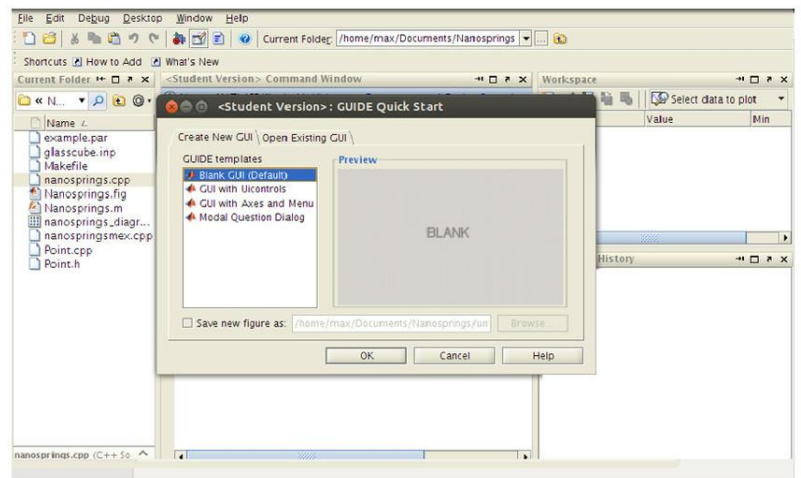


Рисунок 4 Інтерфейс MATLAB

введення команд мови програмування в «Вікно команд», тому для використання програми необхідно знання основних концепцій програмування.

Отже, усі розглянуті програмні інструменти є потужними математичними програмами, які використовуються в наукових та інженерних дослідженнях, а також в освітніх закладах для викладання математики. Вони надають можливості для розв'язання складних математичних задач, візуалізації даних та створення графіків.

### 3. РОЗРОБКА ВЕБЗАСТОСУНКУ

#### 3.1. Загальний опис програми та її реалізації

Як було зазначено вище, найвідоміші існуючі інструменти можуть робити багато дій, через це вони мають досить складний інтерфейс, який може здатися незрозумілим для вчителів та учнів. Саме тому, я розробляю застосунок з простим інтерфейсом, який дозволяє легко вирішувати задачі. Для цього, моя програма має виконувати наступні функції:

- Вводити формули у зручний спосіб: користувач має мати можливість ввести формули, використовуючи клавіатуру або спеціальні кнопки на інтерфейсі програми.
- Відображати формули: після введення формул, програма має відображати їх у зручному для користувача форматі – у вигляді графічного зображення.
- Надавати користувачу можливість виділяти формули або їх частини: користувач має мати можливість виділяти окремі елементи формул для подальшого перетворення.
- Виконувати певні математичні перетворення виділених формул або їх частин: після виділення елементів формул, програма має забезпечувати їх перетворення у зручний для користувача спосіб, так само, як це робиться на папері.

Оскільки було оголошено, що програма має бути зручною у використанні, я вирішила написати застосунок, який буде працювати через браузер, оскільки це дозволить користувачам отримувати доступ до програми з будь-якого місця, де є доступ до інтернету. Я використовуватиму мову JavaScript разом з HTML та CSS для написання мого вебзастосунку, тому що ці технології широко використовуються для розробки вебзастосунків і мають багато корисних функцій та бібліотек. Зокрема, HTML використовується для створення структури та

відображення контенту на сторінці, CSS - для оформлення цього контенту, а JavaScript - для динамічних ефектів та взаємодії з користувачем.

Окрім стандартних можливостей JavaScript, HTML та CSS будуть використані наступні бібліотеки:

**Bootstrap** – є однією з найпопулярніших бібліотек для створення вебінтерфейсів, яка надає набір готових CSS-стилів та JavaScript-компонентів, таких як кнопки, форми, різноманітні меню тощо. Ця бібліотека буде використана для створення красивого та зручного інтерфейсу користувача.

**MathQuill** – це бібліотека для створення візуальних редакторів математичних формул на вебсторінках. Вона дозволяє користувачам вводити математичні вирази, використовуючи звичайний клавіатурний ввід. Результат введення можна отримати у вигляді LaTeX-розмітки.

**MathJax** – це бібліотека для відображення математичних формул на вебсторінках. Вона підтримує відображення математичних формул у всіх браузерах та на всіх пристроях. Ця бібліотека може відображати формули подані у вигляді AsciiMath або LaTeX розмітки.

## 3.2. Опис формули

### 3.2.1. Формальна граматики

За допомогою LaTeX-розмітки можна описати будь-які математичні конструкції. Також її використовують обрані бібліотеки для введення та виведення формул. Тому в створюваній програмі математичні вирази будемо описувати LaTeX-розміткою.

Далі нам потрібно аналізувати обрану розмітку. Для цього опишемо синтаксис LaTeX-розмітки за допомогою Форми Бекуса-Наура (БНФ). Ми будемо

описувати не весь синтаксис LaTeX-розмітки, а тільки його частину, яку будемо використовувати в застосунку, а саме: дробы, тригонометричні функції, нижні індекси, верхні індекси (ступінь), вектори та грецькі та латинські літери.

У додатку А розміщено опис синтаксису LaTeX-розмітки, у вигляді БНФ нотації.

### 3.2.2. Структура даних

У процесі розробки мого застосунку я стикнулася з проблемою зображення математичних формул в мові програмування. Звичайні типи даних, такі як рядки, не здатні адекватно зображати складні формули, що ускладнює їх подальшу обробку та перетворення. Тому для забезпечення зручної взаємодії з формулами, я створила спеціальну структуру даних.

Структура формули складається з декількох типів об'єктів (класів), які відповідають основним частинам формули. В об'єкті існує набір атрибутів, які описують, з чого складається описувана частина формули. Кожному атрибуту може бути присвоєний інший зі створених об'єктів, масив із створених об'єктів, або об'єкти примітивних типів *string* та *number*. Кожній властивості може бути присвоєний лише об'єкт визначеного типу, отже властивості є типізованими.

В Додаток Б описана структура формули. Назва типу об'єкту починається з великої літери. Далі в фігурних дужках йде опис властивостей: спочатку назва властивості, потім через двокрапку тип властивості. Якщо після типу написані квадратні дужки «[]», то властивості має присвоюватись масив, який складається з елементів вказаного типу. Якщо після назви типу об'єкту написано «(Mult)», це означає, що даний тип може бути присвоєним будь-якій властивості, яка має тип Mult. Ділі будуть детальніше описані основні частини даної структури.

Тип *Formula* є основним, в ньому описується вся формула. Його властивість *equalityParts* – це набір виразів, які на письмі поєднуються знаком «=». Якщо структура описує звичайний вираз, то *equalityParts* складається з одного елемента. Якщо вона описує рівняння – то з двох елементів (правої та лівої частини рівняння). Якщо описується послідовність тотожних перетворень виразу, то кількість елементів може бути довільною.

Тип *Expr* описує будь-який вираз. Властивість *content* – це набір доданків, з яких складається описуваний вираз. Наприклад, вираз  $a^2+b^2-2ab$  складається з 3 доданків:  $+a^2$ ,  $+b^2$  та  $-2ab$ , а вираз  $-(a+b)^2$  складається з одного доданка:  $-(a+b)^2$ .

Тип *Term* описує будь-який доданок. Властивість *sign* описує знак доданка, він може бути «+» або «-». На письмі знак доданка ставиться перед множниками, або не ставиться, якщо доданок стоїть на початку виразу та має знак «-». Властивість *content* – це набір множників, які поєднуються операцією множення «\*». Множниками можуть бути вирази (записуються у дужках), числа, дроби, змінні тощо. Наприклад в виразі  $a^2 - 2ab + b^2$  перший та третій доданки мають знак «+», другий має знак «-». Наприклад, доданок  $2a^2(a+b)$  має три множники: 2,  $a^2$  та  $(a+b)$ ; доданок  $(-1.5a)*b$  має два множники:  $(-1.5a)$  та  $b$ ; доданок  $+(-1.5)$  має єдиний множник  $(-1.5)$ , який є виразом.

Тип *Fraction* описує дроби та має дві властивості: *numerator* (числівник) та *denominator* (знаменник). В формальній граматиці та в звичайному записі дробу числівник та знаменник є виразами. Однак, при виконанні основних математичних дій з дробами, таких як множення та ділення, зручніше описувати числівник та знаменник за допомогою набору множників, тобто за допомогою типу *Term*. Якщо числівник та знаменник дробу мають тип *Term*, то операцію множення (ділення) дробів можна описати, як додавання множників числівника та знаменника одного дробу до іншого дробу.



### 3.2.3. Перетворення структури даних

Вище було зазначено, що для введення та зображення формул я буду використовувати бібліотеки MathJax та MathQuill, які працюють з LaTeX-розміткою. Оскільки моя структура даних для представлення формул не може бути використана безпосередньо з цими бібліотеками, я вирішила створити функції, які дозволяють перетворювати мою структуру даних у LaTeX-розмітку та навпаки – отримувати структуру даних з LaTeX-розмітки.

Для отримання структури з LaTeX-розмітки я використовую синтаксичний аналіз. Синтаксичний аналіз виконується методом рекурсивного спуску. Для розпізнання кожного нетермінала граматики (Додаток А) була створена окрема функція, яка відповідає за розпізнавання цього нетермінала та повертає відповідну частину структури даних.

Для того, щоб перетворювати структуру даних у LaTeX, я використовую рекурсію. Для кожного типу об'єкту, з якого складається структура, я створила метод, який перетворює цей тип у LaTeX-формат. Ці методи викликаються рекурсивно для кожного атрибута, який також є об'єктом зі своїм власним методом перетворення в LaTeX. Таким чином, я можу перетворити всю структуру даних у LaTeX, збираючи всі шматочки разом за допомогою рекурсивних викликів.

### 3.3. Дії з формулами

Створювана програма може виконувати наступні дії з формулами: перетворення рівнянь та перетворення виразів. Перетворення рівняння можна виконувати формулами, які мають принаймні дві частини. Такі дії змінюють обидві частини формули (рівняння). Перетворення виразу працюють з виразами

або їх частинами. Отже, після виконання перетворення виразу змінюється лише одна частина формули.

Формула може складатись більше ніж з двох частин, наприклад, коли формула зображує послідовність перетворень виразу:

$$F = mg = \rho Vg$$

З такою формулою також можна робити перетворення рівняння, в цьому випадку перетворення рівняння будуть використовувати першу та останню частини формули. Наприклад, з вищеописаної формули можна виразити змінну  $V$ , в результаті отримаємо:

$$V = \frac{F}{\rho g}$$

Зауважимо, що з цієї формули не можна виразити змінну  $m$ , оскільки вона не належить ні першій, ні останній частині формули.

### 3.3.1. Перетворення рівняння

**Виразити доданки:** обрані доданки переносяться в ліву частину рівняння, усі інші – в праву.

$$ma = -2F + mg \rightarrow 2F = mg - ma$$

**Виразити множник:** виражається доданок, який містить обраний множник, потім за пропорцією виражається обраний множник.

$$E = \frac{mv^2}{2} \rightarrow v^2 = \frac{2E}{m}$$

$$ma = F \cos \alpha - \mu mg \rightarrow F = \frac{ma + \mu mg}{\cos \alpha}$$

**Помножити рівняння:** кожен доданок рівняння помножається на деякий обраний множник.

$$\frac{x}{y} + \frac{y}{x} = 2 \text{ помножити на } xy \rightarrow x^2 + y^2 = 2xy$$

**Додати рівняння:** праві та ліві частини обраних рівнянь почленно додаються.

$$\begin{array}{l} x + 2y = 7 \\ 3x - 2y = 3 \end{array} \rightarrow 4x = 10$$

**Відняти рівняння:** частини одного рівняння почленно віднімаються від частин іншого рівняння.

$$\begin{array}{l} 2x + 2y = 4 \\ 2x - y = 3 \end{array} \rightarrow 3y = 1$$

**Ділити рівняння:** частини одного рівняння діляться на частини іншого.

$$\begin{array}{l} a_1 = \frac{F}{m_1} \\ a_2 = \frac{F}{m_2} \end{array} \rightarrow \frac{a}{a_2} = \frac{m_2}{m_1}$$

### 3.3.2. Перетворення виразів

**Підстановка:** для виконання цієї дії необхідно обрати рівняння, у якого в лівій частині стоїть лише один множник, потім в іншій формулі обрати множник, рівний множнику лівої частини обраного рівняння. В процесі виконання дії цей множник заміниться на праву частину рівняння.

$$\frac{v^2}{2} = \frac{v_x^2 + v_y^2}{2} \rightarrow \frac{m(v_x^2 + v_y^2)}{2}$$

**Розкрити дужки:** для виконання цієї дії необхідно обрати множник, який є виразом в дужках. В результаті вираз всередині дужок помножається на те, що стоїть перед або поза дужками.

$$a(b + c) \rightarrow ab + ac$$

**Винести множник за дужки:** для виконання цієї дії необхідно обрати декілька доданків, які знаходяться в одному виразі, та деякий множник. В результаті обрані доданки замінюються доданком, який містить обраний множник та вираз в дужках. Всередині дужок знаходяться обрані доданки, поділені на обраний множник.

$$ab + ac \text{ винести } a \rightarrow a(b + c)$$

$$x^2 + y^2 \text{ винести } xy \rightarrow xy \left( \frac{x}{y} + \frac{y}{x} \right)$$

**Звести до спільного знаменника.** Обрані доданки, які стоять в одній частині, зводяться до спільного знаменника.

$$\frac{a}{b+1} + \frac{b}{a+1} \rightarrow \frac{a(a+1) + b(a+1)}{(a+1)(b+1)}$$

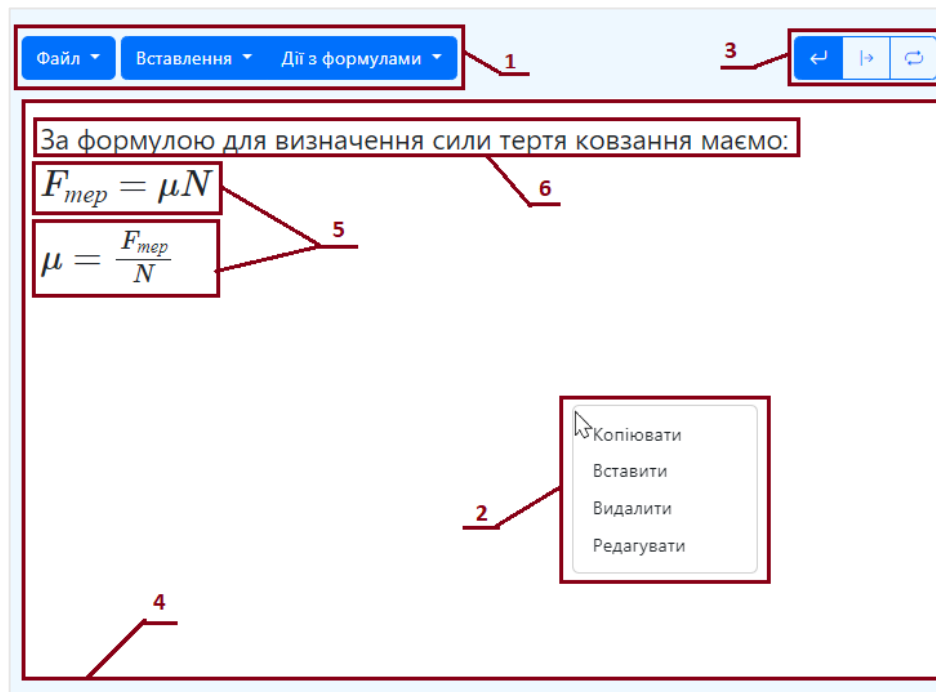
### 3.4. Інструкція для користувача

Використовувати програму можна за посиланням:

<https://interactive-formulas.netlify.app>.

### 3.4.1. Елементи управління програмою

Пояснимо основні частини інтерфейсу програми:



1. Головне меню. Через нього можливий доступ до основних дій програми.
2. Контекстне меню. Викликається натисканням правої кнопки миші.
3. Режим додавання нової частини формули. Детальніше в розділі 3.4.4
4. Інтерактивне поле. В ньому відбувається розв'язання задачі.
5. Інтерактивна формула. Частина інтерактивного поля. З нею можливо виконувати перетворення та взаємодіяти.
6. Текст. Частина інтерактивного поля, яка пояснює інтерактивну формулу.

### 3.4.2. Додавання елементів до інтерактивного поля

Додавання елементів до інтерактивного здійснюється через натискання кнопки головного меню «Вставлення». Потім потрібно обрати «тест» або «формула» в залежності від того, що ви хочете додати.

Розглянемо інтерфейс додавання формули. В поле введення вводимо необхідну формулу. Введення можливо здійснювати як за допомогою звичайної клавіатури, так і за допомогою віртуальної клавіатури, що знаходиться внизу сторінки. Зауважимо, що вводити можна тільки ті символи, які присутні в віртуальній клавіатурі, інакше формула не зможе бути доданою до інтерактивного поля. Інтерактивна клавіатура містить 3 вкладки: з основними символами та шаблонами, з грецькими літерами та з латинськими літерами.

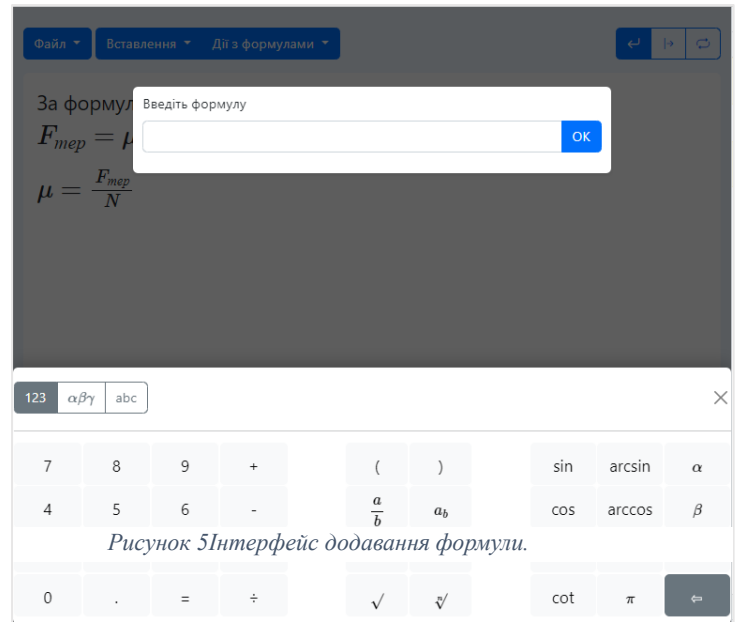


Рисунок 5 Інтерфейс додавання формули.

Розглянемо інтерфейс додавання тексту. Текст, який нам необхідно додати, вводиться у нижнє поле. Також ми можемо додати формулу до тексту. Це можливо зробити за допомогою введення необхідної формули у верхнє поле та натискання кнопки «Вставити ↴». Формули в тексті можуть містити будь-які символи, але з ними не можливо взаємодіяти окремо від тексту.

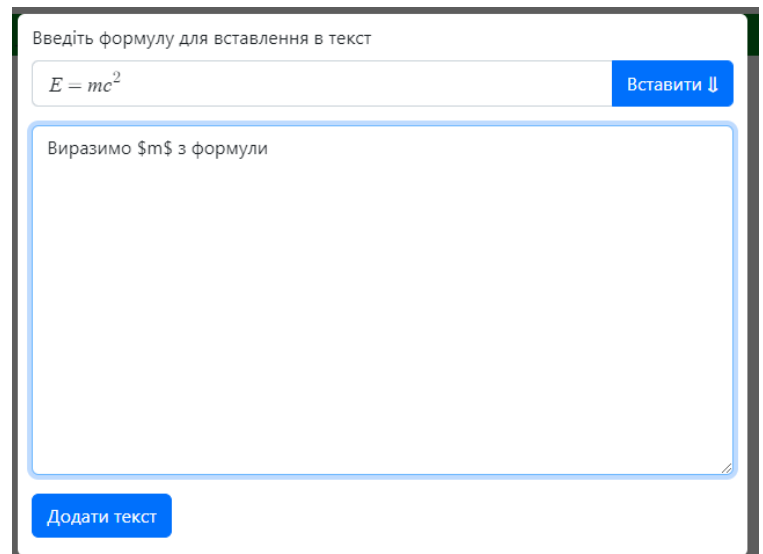


Рисунок 6 Інтерфейс додавання тексту.

### 3.4.3. Дії з формулами

Для виконання перетворень формули необхідно виділити частини формули, з якими буде виконуватись дія. Потім потрібно натиснути кнопку головного меню «Дії з формулами» та обрати необхідне перетворення з випадаючого меню.

В таблиці нижче показано, які частини формули користувач може виділити.

Назва	Приклад	Як виділити
Формула	$x^2 - 7x - 18 = 0$	Клацніть праворуч від формули, яку необхідно вибрати.
Доданок	$x^2 - 7x - 18 = 0$	Клацніть по доданку, який необхідно вибрати.
Множник	$a + \frac{5+b}{2a}$ $a + \frac{5+b}{2\sqrt{a}}$	Двічі клацніть по множнику, який необхідно вибрати.

Для виділення декількох об'єктів необхідно затиснути клавішу «shift».

Зауважимо, що користувач не може виділяти доданки, які містяться всередині множника. Але для того, щоб взаємодіяти з частинами множника, який є виразом у дужках або знаменником чи чисельником дробу, необхідно виділити цей множник та застосувати дію «Сфокусувати».

Розглянемо, які дії може виконувати програма.

Назва дії	Що необхідно виділити	Приклад	
		виділили	отримали
«Виразити»	1+ доданків	$ab + df = ac + de$	$ab - ac = de - df$
	1 множник	$15y - 6x = -15$	$x = \frac{15 + 15y}{6}$

«Помножити рівняння»	1 формула	$\frac{a^2}{b^2} + \frac{a}{b} + 1 = 0$ <i>Введення <math>b^2</math></i>	$a^2 + ab + b^2 = 0$
«Додати рівняння»	2+ формули	$15x + y = 5$ $x - y = 23$	$16x = 28$
«Відняти рівняння»	2 формули	$3a + b = 1$ $2a + b = 3$	$-a = 2$
«Ділити рівняння»	2 формули	$a(b + c) = d$ $a(a + c) = d$	$\frac{b + c}{a + c} = 1$
«Підставити»	1 множник (виражений <sup>1</sup> ), 1 множник (рівний першому)	$x = \frac{15}{y}$ $z = 7y + xy$	$z = 7y + 15$
«Звести до спільного знаменники»	2+ множника в одному виразі	$y = \frac{1}{5x} + \frac{3}{10}$	$y = \frac{3x+2}{10x}$
«Розкрити дужки»	1 множник (вираз в дужках)	$y = 5x(x + 5)$	$y = 5x^2 + 5^2x$
«Винести множник за дужки»	2+ множника	$y = 3x - ax$ <i>Введення <math>x</math></i>	$y = x(3 - a)$

<sup>1</sup> Виражений множник – єдиний множник у правій частині формули



«Підставити усі змінні»	1 формула з вираженим множником	$a = 5$ $b = -1$ $c = \frac{ab}{a+b}$	$c = \frac{5(-1)}{5+(-1)}$
«Розрахувати»	1 формула з вираженим множником	$c = \frac{5(-1)}{5+(-1)}$	$c = -1.25$

#### 3.4.4. Режим додавання нової частини

У перетвореннях, в яких змінюється лише одна частина формули, існують різні способи отримання нової формули зі зміненою частиною. Те, як буде отримуватись нова формула, визначає «Режим додавання нової частини» (див Рисунок 7)

1. «З нового рядка» В інтерактивне поле додається нова формула, яка містить частину, яка не змінювалась в процесі виконання дії, та перетворену частину.



Рисунок 7

2. «Додавання в кінець». Змінена частина додається в кінець формули.

3. «Заміна». Частина формули, в якій виконується перетворення, замінюється перетвореною частиною.

Приклади виконання дії «Розкрити дужки» в залежності від режиму додавання нової частини.

Початкова формула:	$a = b(c + d)$
--------------------	----------------

Режим	Результат
«З нового рядка»	$a = b(c + d)$ $a = bc + bd$
«Додавання в кінець»	$a = b(c + d) = bc + bd$
«Заміна»	$a = bc + bd$

### 3.4.5. Інші дії програми

Розглянемо дії контекстного меню:

**«Копіювати».** Копіює виділений текст або LaTeX віділеної формули до буферу обміну.

**«Вставити».** Додає текст з буферу обміну або попередньо скопійовану формулу до інтерактивного поля.

**«Видалити».** Видаляє з інтерактивного поля виділені тексти та формули.

**«Редагувати».** Дозволяє редагувати виділену формулу або текст.

Дії «Копіювати» та «Вставити» можна робити без використання контекстного меню за допомогою комбінації клавіш «ctrl + c» та «ctrl + v».

Для виконання дій пов'язаних зі збереженням або відкриттям розв'язку необхідно натиснути на кнопку головного меню «Файл», у випадяючому меню можна побачити список доступних дій:

**«Зберегти як зображення».** Дозволяє зберегти розв'язок у вигляді зображення з розширенням .png.

**«Зберегти як LaTeX».** Зберігає розв’язок задачі у вигляді LaTeX розмітки.

**«Відкрити».** Дозволяє відкрити попередньо збережений як LaTeX розв’язок або будь-який файл з LaTeX розміткою.

## 4. ЗАСТОСУВАННЯ РЕЗУЛЬТАТІВ РОЗРОБКИ

В сучасному світі, де дистанційне навчання стає все більш поширеним, зручні та ефективні інструменти для викладачів та учнів є дуже важливими. Проте, часто учням та викладачам складно працювати з математичними формулами через проблеми з почерком, незручність у показі процесу розв'язання завдання та інші причини. За допомогою створеного застосунку можна вирішити ці проблеми.

### 4.1. Застосування в математиці

Уявимо, що учню на уроці математики необхідно вирішити наступну систему рівнянь:

$$\begin{cases} 7x + 3y = 6 \\ 2x - 8y = 19 \end{cases}$$

Далі буде надана інструкція до того, як можна вирішити цю задачу, використовуючи створений застосунок:

1. Введемо рівняння. Для цього натискаємо «Вставлення» > «Формула». В поле введення введемо перше рівняння, та натискаємо «ОК». Так само вводимо друге рівняння.
2. Виразимо змінну «у» з першого рівняння. Для цього виділимо множник «у» в першій формулі, двічі клацнувши по ньому. Потім натиснемо «Дії з формулами» > «Виразити».
3. Підставимо отриманий вираз у другу формулу. Для цього виділимо множник «у» в новій формулі, а також в другій введеній

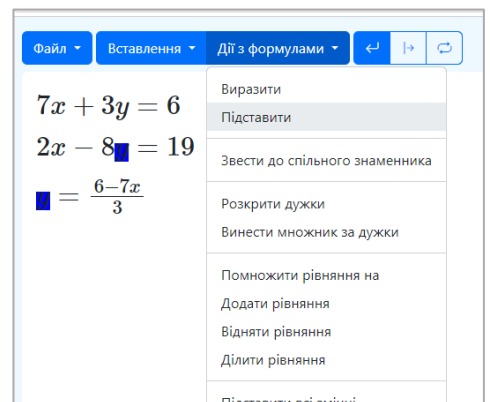


Рисунок 8

формулі, затиснувши shift. Потім натиснемо «Дії з формулами» > «Підставити». (Рисунок 8)

4. Помножимо отримане рівняння на 3. Для цього виділимо формулу, натиснувши праворуч від неї. Потім натиснемо «Дії з формулами» > «помножити рівняння», введемо у поле введення «3», та натиснемо «ОК».

5. Розкриємо дужки в отриманому рівнянні. Для цього виділимо множник, що містить дужку та натиснемо «Дії з формулами» > «Розкрити дужки».

6. Аналогічно кроку 2 виразимо з отриманої формули «x»

7. Аналогічно кроку 3 підставимо отриманий вираз в рівняння з вираженою змінною у.

Система розв'язана (Рисунок 9) .

Якщо учню потрібно надіслати цей розв'язок, він може зберегти його натиснувши «Файл» > «Зберегти як зображення».

$$\begin{aligned} 7x + 3y &= 6 \\ 2x - 8y &= 19 \\ y &= \frac{6-7x}{3} \\ 2x - \frac{8(6-7x)}{3} &= 19 \\ 6x - 8(6-7x) &= 57 \\ 62x - 48 &= 57 \\ x &= \frac{105}{62} \\ y &= -\frac{121}{62} \end{aligned}$$

Рисунок 9. Повний розв'язок

Отже, за допомогою створеної програми ми змогли вирішити математичну задачу в електронному вигляді. Крім того, «технічні» дії, такі як калькуляція, були виконані програмою автоматично, що дозволило учню більше сконцентруватись на алгоритмі вирішення задачі.

## 4.2. Застосування в фізиці

Уявимо, що вчителю необхідно показати учням процес розв'язати наступну задачу:

*«Мідна каструля має масу 0.5 кг. Якщо каструлю такого самого розміру виготовити зі сталі, яка в неї буде маса? (Густина міді - 8900 кг/м<sup>3</sup>, густина сталі - 7800 кг/м<sup>3</sup>)»*

Для цього вчителю достатньо продемонструвати екран з застосунком та виконати в ньому наступні дії:

1. Спочатку для зручності розв'язання можемо скопіювати текст та додати його до інтерактивного поля через натискання комбінації клавіш `ctrl + v`, або через команду контекстного меню «Вставити».
2. Введемо дано. Це можна зробити через натискання «Вставлення» > «Формула». Символ « $\rho$ » та нижні індекси можна ввести за допомогою віртуальної клавіатури.
3. Для того, щоб відділити частину з тим, що дано за умовою задачі, та тим, що потрібно знайти, вставимо розділювач. Для цього натиснемо «Вставлення» > «Текст» та введемо у нижнє поле декілька знаків «-» та натиснемо «Додати текст» (Рисунок 10).

Мідна каструля має масу 0.5 кг. Якщо каструлю такого самого розміру виготовити зі сталі, яка в неї буде маса? (Густина міді - 8900 кг/м<sup>3</sup>, густина сталі - 7800 кг/м<sup>3</sup>)

$m_1 = 0.5$   
 $\rho_1 = 8900$   
 $\rho_2 = 7800$   
 -----

Рисунок 10

4. Додаймо текст, в якому зазначимо, що нам потрібно знайти. Для цього натиснемо «Вставлення» > «Текст». Для того, щоб додати в текст  $m_2$ , введемо потрібні символи в верхнє поле введення та натиснемо «Вставити ↵» (Рисунок 11). Потім закінчуємо набір тексту та натискаємо «Додати текст».

Введіть формулу для вставлення в текст

$m_2 =$

Рисунок 11

5. Далі можемо скопіювати розділювач, який ми додали на кроці 3, за допомогою команди `ctrl + c` або функції контекстного меню «Копіювати», та вставити його.
6. Далі додаймо формули  $m_1 = V\rho_1$  та  $m_2 = V\rho_2$ . Першу формулу можна ввести через команду «Вставлення» > «Формула». Другу формулу можна додати аналогічно першій, або можна скопіювати першу формулу, потім за

допомогою команди контекстного меню «Редагувати» замінити в цій формулі нижні індекси.

7. Поділимо додані на попередньому кроці рівняння. Для цього виділимо їх, затиснувши shift, та натиснемо «Дії з формулами» > «Ділити рівняння».

8. З отриманої формули виразимо змінну  $m_2$ . Для цього виділимо її в останній формулі та натиснемо «Дії з формулами» > «Виразити».

9. Потім підставимо в отриманий вираз змінні з дано. Для цього виділимо формулу, в яку необхідно підставити змінні, та натиснемо «Дії з формулами» > «Підставити всі змінні».

10. Підрахуємо результат. Для того, щоб результат розрахунку з'явився не в новій формулі, а додався до формули, в якій відбуваються перетворення, виберемо режим додавання нової частини



Рисунок 12

«Додавання в кінець» (Рисунок 12). Потім виділимо формулу, в якій необхідно розрахувати праву частину, та натиснемо «Дії з формулами» > «Розрахувати».

*Задача вирішена (Рисунок 13)*

Отже, за допомогою створеної програми вчитель зміг продемонструвати процес розв'язання фізичної задачі. Крім того, програма змогла полегшити процес розв'язання через автоматичну підстановку значення змінних та калькуляцію відповіді.

Мідна каструля має масу 0.5 кг. Якщо каструлю такого самого розміру виготовити зі сталі, яка в неї буде маса?  
(Густина міді - 8900 кг/м<sup>3</sup>, густина сталі - 7800 кг/м<sup>3</sup>)

$$m_1 = 0.5$$

$$\rho_1 = 8900$$

$$\rho_2 = 7800$$

$$m_2 = ?$$

$$m_1 = V \rho_1$$

$$m_2 = V \rho_2$$

$$\frac{m_1}{m_2} = \frac{\rho_1}{\rho_2}$$

$$m_2 = \frac{\rho_2 m_1}{\rho_1}$$

$$m_2 = \frac{7800 \cdot 0.5}{8900} = 0.438$$

Рисунок 13. Повний розв'язок

## Висновок

За результатами виконання курсової роботи можна зробити висновок, що поставлені завдання були успішно виконані.

Спочатку були вивчені теоретичні основи формальних мов і граматик, що дозволило зрозуміти принципи побудови формул і створення їх відповідно до визначених граматик. Потім я ознайомилась з існуючими інструментами для роботи з формулами, що дало змогу виявити їх переваги та недоліки.

На основі цього було розроблено вебзастосунок, який допомагає школярам та вчителям вирішувати будь-які задачі, пов'язані з математичними формулами. Застосунок дозволяє виконувати перетворення аналогічні тим, що ми робимо на папері, але в електронному вигляді, що забезпечує зручність та ефективність його використання. Також були розглянуті можливості та перспективи застосування результатів розробки. Зокрема, застосунок може бути корисним як для школярів, які можуть використовувати його для вирішення задач, так і для вчителів, які можуть використовувати його для підготовки уроків.

Отже, результатом цієї курсової є зручний та ефективний вебзастосунок, який допомагає вирішувати задачі, пов'язані з математичними формулами, а також може мати широке застосування у навчанні.



## Список літератури

1. Meduna A. Formal languages and computation. Models and their applications / Alexander Meduna – Boca Raton, London, New York: CRC Press, 2014. С. 13-25
2. Компиляторы. принципы, технологии и инструментарий / А.Ахо, М. Лам, Р. Сети, Д. Ульман. – Москва, Санкт-Петербург, Киев: Вильямс, 2008. С. 251-286.
3. Maple - The Essential Tool for Mathematics - Maplesoft [Електронний ресурс] – Режим доступу до ресурсу: <https://www.maplesoft.com/products/Maple>.
4. Wolfram Mathematica: Modern Technical Computing [Електронний ресурс] – Режим доступу до ресурсу: <https://www.wolfram.com/mathematica>.
5. MATLAB - MathWorks [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mathworks.com/products/matlab.html>.
6. MathJax | Beautiful math in all browsers. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mathjax.org>.
7. MathQuill: Easily type math into your webapp [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.mathquill.com/en/latest>.
8. Bootstrap. The most popular HTML, CSS, and JS library in the world. [Електронний ресурс] – Режим доступу до ресурсу: <https://getbootstrap.com>.

## Додатки

## Додаток А

Нотація БНФ, яка описує синтаксис LaTeX

```

<formula> ::= <expr> = <formula> | <expr>
<expr> ::= <term><expr> | <term>
<term> ::= <pm><mult-group>
<pm> ::= + | - | <empty>
<mult-group> ::= <mult>*<mult-group> | <mult><mult-group> | <mult>
<mult> ::= <frac> | <function> | <number> | <sqrt> | <exponent> | <variable> |
<vector> | (<expr>)
<fraction> ::= \frac{<expr>}{<expr>}
<function> ::= <functionName>(<expr>)
<functionName> ::= \sin | \cos | \tan | \ctg | \arcsin | \arccos | \arctan | \arcctg
<number> ::= <integer> | <integer>.<integer>
<integer> ::= <digit> | <digit><integer>
<sqrt> ::= \sqrt{<expr>} | \sqrt[<expr>]{<expr>}
<exponent> ::= <mult>^{<expr>}
<vector> ::= \vec{<variable>}
<variable> ::= <var-name><sub><primes>
<primes> ::= '<primes> | <empty>
<sub> ::= _{<expr>} | <empty>
<var-name> ::= <latin> | <greek>

```

В цій нотації використовуються наступні невизначені нетермінали:

<empty> – позначає порожній рядок.

<greek> – позначає будь-яку літеру грецького алфівіту у LaTeX запису (“\” + назва літери). Наприклад:  $\backslash alpha$  ( $\alpha$ ),  $\backslash beta$  ( $\beta$ ),  $\backslash Sigma$  ( $\Sigma$ ),  $\backslash Omega$  ( $\Omega$ ).

<latin> – позначає будь-яку літеру латинського алфавіту в нижньому або верхньому регістрі.

<digit> – позначає будь-яку цифрку від 0 до 9

## Додаток Б

## Структура формули

Formula{ equality_parts: Expr[] }	Number(Mult){ value: number }
Expr{ content: Term[] }	Sqrt(Mult){ root: Block content: Mult }
Term{ sign: "+"   "-" content: Mult[] }	Exponent(Mult){ exponent: Block base: Mult }
Fraction(Mult){ numerator: Term denominator: Term }	Variable(Mult){ name: string index: string vector: bool primeCount: number; }
Function(Mult){ name: string content: Block }	