Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

«Побудова графіків неявно заданих функцій, побудова кількох графіків на малюнку і додавання тексту мовою Java»

Керівник курсової роботи
Завідувач кафедри, професор,
доктор фізико-математичних наук
Малашонок Г. І.

_____
(підпис)
"____"_____2024 р.

Виконала студентка 3 р. н.
Прокопеня П. С.
"____"_____2024 р.

Київ 2024
Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра інформатики факультету інформатики

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студентки Прокопені Поліни Сергіївни

3-го курсу факультету інформатики

ТЕМА: Побудова графіків неявно заданих функцій, побудова кількох графіків на малюнку і додавання тексту мовою Java

Вихідні дані:

Зміст ТЧ до курсової роботи

Вступ

Анотація

1. Аналіз предметної області
2. Вибір технологій та засобів для розробки
3. Побудова графіків неявно заданих функцій
4. Побудова кількох графіків на малюнку та додавання тексту до графіків

Висновки

Список використаних джерел

Календарний план виконання курсової роботи

Тема: Побудова графіків неявно заданих

функцій, побудова кількох графіків на малюнку і додавання тексту мовою Java

Календарний план виконання роботи:

№ п/п	Назва етапу дипломного проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу	Жовтень 2023 р.	
2.	Аналіз предметної області	Листопад 2023 р.	
3.	Вибір бібліотек та розробка логіки для імплементації функціоналу	Січень-лютий 2024 р.	
4.	Розробка кодової частини	Березень 2024 р.	
5.	Написання текстової частини курсової	Квітень 2024 р.	
6.	Оформлення презентації для захисту	Травень 2024 р.	
7.	Здача роботи для перевірки на плагіат	14 травня 2024 р.	
8.	Захист курсової роботи	21 травня 2024 р.	

Студентка: Прокопеня П. С.

Керівник: Малашонок Г. І.

“_____” _____

Зміст

Вступ

У динамічних галузях обчислювальної математики та графічної візуалізації Java залишається ключовою мовою, визнаною завдяки своїм надійним стандартним бібліотекам, крос-платформним можливостям та широкому застосуванню як в академічному, так і в

промисловому середовищі. Ця курсова робота присвячена вдосконаленню графічних функцій Java додатку, з акцентом на складнощі та рішення, пов'язані з побудовою графіків неявно визначених функцій, відображенням декількох графіків на одному малюнку та вбудовуванням текстових анотацій у графіки.
Ці функції є надзвичайно важливими та використовуються у багатьох сферах - від академічних досліджень та освітніх інструментів до професійного аналізу даних та інженерних презентацій. Наприклад, в освітньому середовищі викладачі часто використовують графічні інструменти для пояснення складних функцій у таких предметах, як математика чи фізика, наприклад, для візуалізації кіл або гіпербол для пояснення геометричних та алгебраїчних понять. Дослідження в "Journal of Science Education and Technology" свідчить, що студенти, які використовують інтерактивні графічні інструменти, значно краще розуміють математичні концепції, ніж ті, хто їх не використовує.

Неявно задані функції, ілюструють складні взаємозв'язки і відображення цих функцій з точністю вимагає складних графічних алгоритмів і глибокого розуміння чисельних методів.
Можливість одночасного відображення декількох графіків на одній координатній площині полегшує порівняльний аналіз й покращує взаємодію з користувачем, роблячи візуальне дослідження більш інтуїтивно зрозумілим і глибоким.
Інтеграція текстових анотацій у графіки значно розширює інформацію, що передається, дозволяючи безпосередньо передавати інформацію про дані, математичні залежності або конкретні примітки у визначених точках або областях графіку.

Ця курсова робота заглиблюється в імплементацію цих розширених можливостей додатку для побудови графіків на Java,

використовуючи такі бібліотеки, як exp4j і JavaFX, а також інші інструменти на основі Java.

Її метою є реалізація зазначеного функціоналу, яке не лише покращить візуальне представлення даних, але й спростить взаємодію з користувачем та підвищить загальну корисність додатку. Робота складається з чотирьох розділів.

Перший включає в себе аналіз предметної області, а саме огляд додатків для побудови математичних графіків з фокусом на імплементацію в них розглянутих у курсовій роботі можливостей, а також визначено задачу дослідження.

У другому розділі буде проведений опис та обґрунтований вибір інструментів для розробки функціоналу побудови графіків.

Третій розділ присвячений дослідженню основних алгоритмів побудови неявно визначених функцій та імплементація обраного алгоритму мовою Java.

У четвертому розділі міститься імплементація побудови декількох графіків на одній координатній площині та додавання текстових приміток до них.

Анотація

Курсова робота присвячена розробці функціоналу побудови графіків неявно заданих функцій, побудови декількох графіків на малюнку та додавання тексту до графіків для вже існуючого веб-застосунку Math Patner за допомогою Java та бібліотеки exp4j - для спрощення оцінки математичних виразів.

1. Аналіз предметної області

1.1 . Огляд додатків для побудови математичних графіків

Досліджуючи веб-додатки, призначені для побудови математичних графіків, можна знайти безліч різноманітних інструментів, кожен з яких оснащений функціями, пристосованими для покращення графічного представлення складних математичних функцій. Ці інструменти часто використовують такі функції, як побудова неявно заданих функцій, керування кількома графіками на одній сторінці та включення текстових анотацій для забезпечення ясності та контексту графічних даних, що відображаються на екрані.

Серед відомих програм - Desmos, зручна платформа, відома своїм

інтуїтивно зрозумілим інтерфейсом. Desmos спеціалізується на побудові графіків неявних функцій, дозволяючи користувачам вводити рівняння у форматі $f(x, y) = c$. Ця можливість поєднується з можливістю одночасного відображення декількох графіків, які користувачі можуть розрізняти за допомогою кольорового кодування. Це полегшує візуальне порівняння різних рівнянь. Крім того, Desmos пропонує функціонал для додавання текстових анотацій безпосередньо на графік, що виявляється безцінним для позначення певних точок, осей або для вбудовування пояснень у сам графік.

GeoGebra - це ще один потужний варіант, зокрема, відомий своєю ефективністю в роботі з неявними виразами з можливістю 2D і 3D візуалізації. Подібно до Desmos, GeoGebra дозволяє створювати кілька графіків на одному полотні, таким чином підтримуючи складні порівняльні дослідження і накладання. Платформа також дозволяє широко кастомізувати текстові анотації, включаючи коригування розміру, положення і вирівнювання відносно побудованих функцій, що підвищує інформативність графіків.

Wolfram Alpha, відомий своєю обчислювальною потужністю, також пропонує можливості для побудови графіків неявних функцій. Він інтерпретує вхідні формули і генерує відповідні графіки, хоча загалом він більше орієнтований на створення результатів одного конкретного запиту. Незважаючи на це, він може відображати кілька графіків на основі багаторівневих запитів, що дозволяє проводити порівняльний аналіз. Текстові анотації у Wolfram Alpha можливі, але вони більш обмежені порівняно з більш спеціалізованими графічними інструментами.

Отже, як ми можемо побачити, подібний функціонал користується

попитом і є імплементованим у багатьох популярних веб-застосунках.

1.2. Огляд додатку Math Partner

Math Partner - це комплексна хмарна система математичних обчислень, відома як MathPartner, розміщена на базі Києво-Могилянської академії (http://mathpartner.ukma.edu.ua). Ця платформа надає можливості символьних обчислень і призначена для вдосконалення навчального процесу у вищих навчальних закладах за допомогою технології хмарних обчислень. Вона дозволяє користувачам виконувати складні математичні розрахунки та візуалізувати результати безпосередньо через сучасний веб-браузер без необхідності встановлення додаткового програмного забезпечення.

MathPartner є однією з перших хмарних систем такого типу, що з'явилася близько 2011 року, і стоїть поряд з іншими системами символьних обчислень SaaS, такими як Sage, Wolfram Alpha та Wolfram Cloud. Ядро MathPartner побудовано на різноманітних технологіях, включаючи специфічну для системи мову, відому як Mathpar. Ця мова полегшує не лише обчислювальні задачі, але й підготовку та проведення лекцій, що робить її невід'ємним інструментом для академічного середовища.

Побудований переважно з використанням Java Spring Boot, Math Partner використовує надійність екосистеми Java для забезпечення можливості розширення та ефективної роботи серверної частини, що є критично важливим для виконання завдань з інтенсивними обчисленнями, таких як математичні побудови в реальному часі та чисельні розрахунки. Проект включає різні технології, в тому числі інтерфейс передачі повідомлень (MPI) для паралельних і

розподілених обчислень, що дозволяє ефективно виконувати великомасштабні математичні операції та симуляції.

Можливості платформи поширюються на динамічну побудову 2D і 3D графіків, маніпуляції з алгебраїчними виразами та виконання складних математичних алгоритмів. Math Partner також діє як освітній інструмент, надаючи ресурси та модулі, спеціально розроблені для цілей навчання та викладання, які включають посібники користувача кількома мовами та навчальні приклади коду на Java. Його структура демонструє модульний підхід, з окремими каталогами і скомпільованими класами для різних функціональних можливостей, пропонуючи дизайн, який підтримує легке налаштування і розширення.

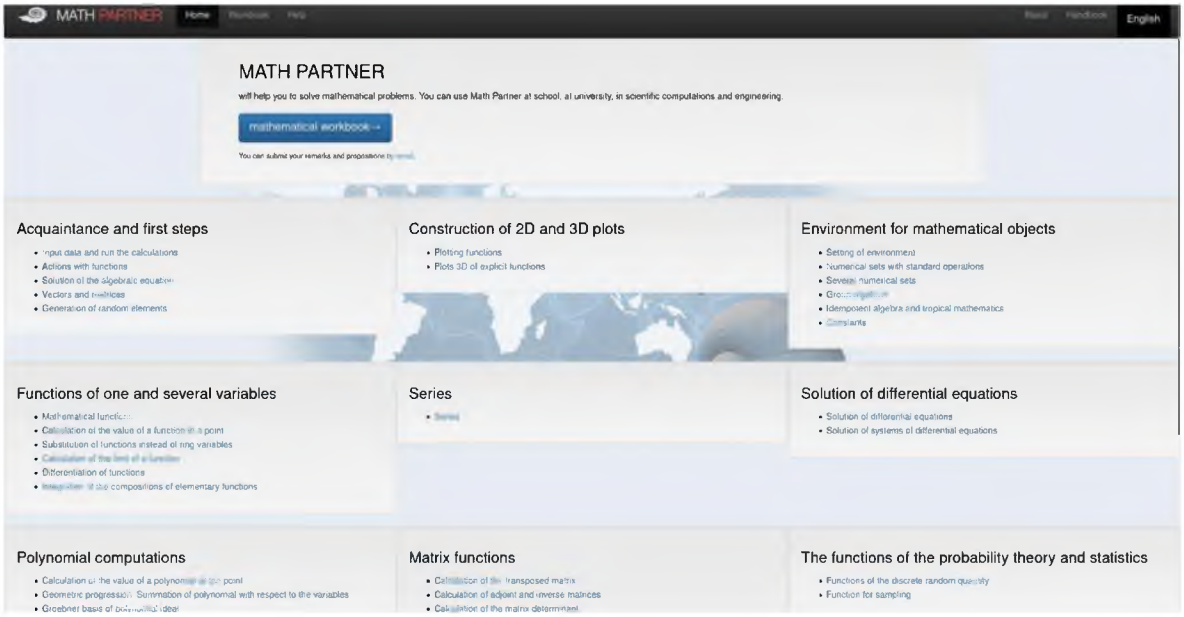

Рис 1.2.1 - Math Partner

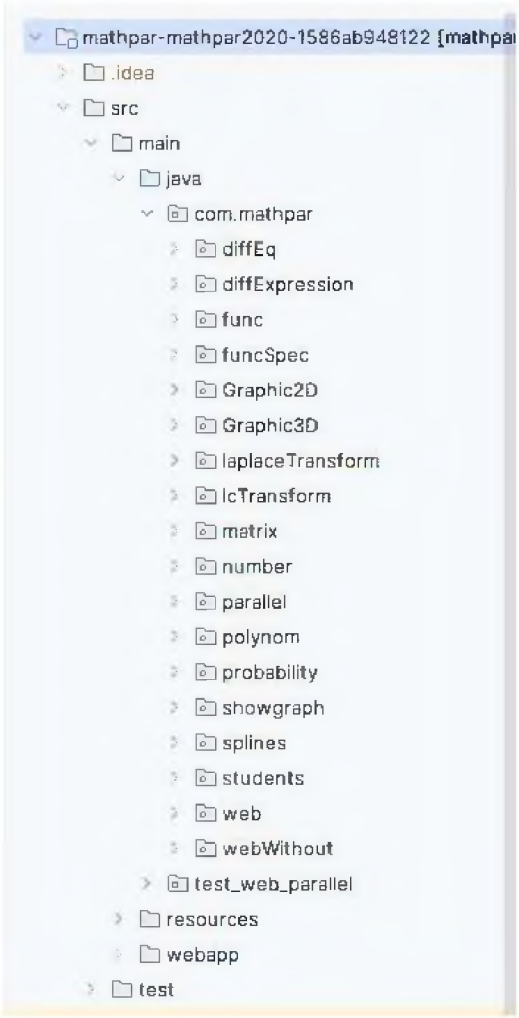

Рис 1.2.2 - струкутра проєкту Math Partner

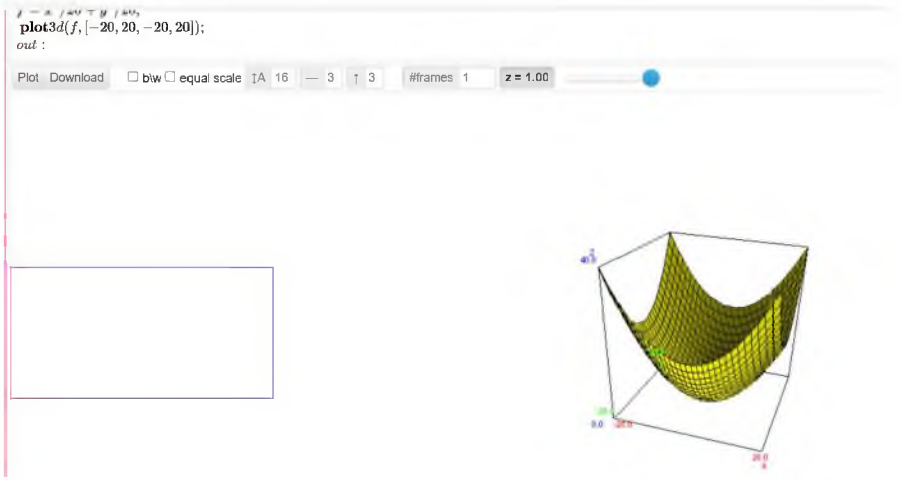

Рис 1.2.3 - приклад побудови графіку за допомогою Math Partner

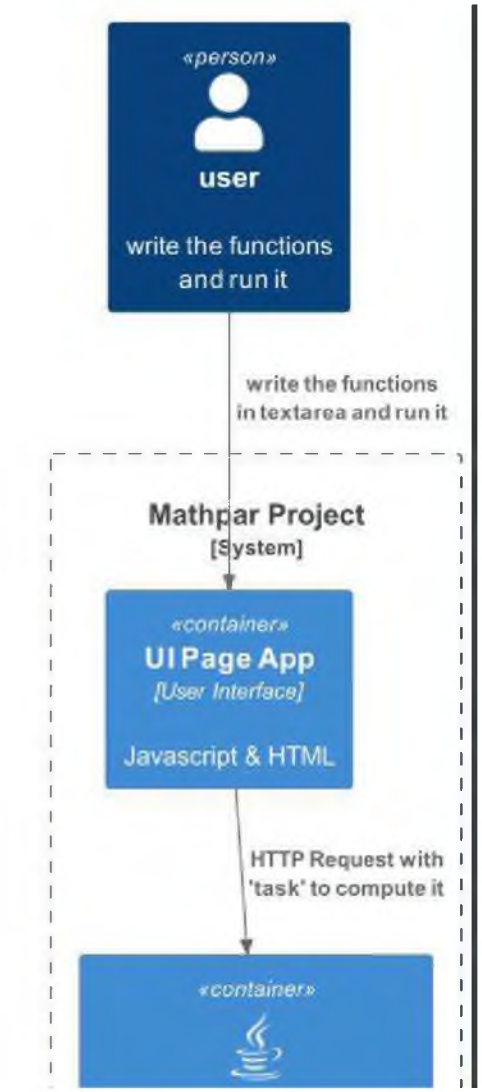

Рис 1.2.4 - структура роботи Math Partner

1.3. Постановка задачі

Враховуючи мету розширення можливостей додатку для візуалізації математичних функцій та представлень даних у мові Java, я можу сформулювати завдання, поставлене переді мною як розробницею, та функції, які буде виконувати цей додаток:

- Реалізація побудови графіків неявно заданих функцій: Розробити модуль, який може інтерпретувати та будувати графіки рівнянь, заданих у неявній формі, використовуючи відповідні числові та графічні алгоритми для точного відображення форм та перетинів цих функцій на декартовій площині.

- Полегшення роботи з декількома графіками на одному малюнку: Реалізація функціоналу, який дозволяє розміщувати кілька графічних панелей або шарів в одному графічному інтерфейсі користувача (GUI), що дає змогу одночасно відображати кілька математичних функцій або наборів даних для порівняльного аналізу та покращеної взаємодії.

- Інтеграція текстових анотацій до графіків: Реалізувати функціонал для додавання описового тексту безпосередньо на графіки, що полегшить маркування ключових моментів, анотацій формул або пояснень, безпосередньо пов'язаних з конкретними частинами графіка, підвищуючи навчальну та комунікативну цінність побудованих даних.

2. Вибір технологій та засобів для розробки

2.1. Загальний огляд Java як мови для написання графічних застосунків

Широке застосування Java для розробки графічних додатків пояснюється її надійними функціями, відмінними крос-платформними можливостями та повним набором бібліотек. Ось чому Java є найкращим вибором для таких додатків:

1) Крос-платформна сумісність: Філософія мови Java "Write once, run anywhere" (WORA) робить її ідеальною для графічних додатків, призначених для роботи на різних системах. Це можливо завдяки віртуальній машині Java (JVM), яка дозволяє запускати Java-додатки на будь-якому пристрої з встановленою JVM, роблячи

додаток незалежним від обладнання та операційних систем.

2) Комплексні API для графічних інтерфейсів: Багатий набір інтерфейсів API Java, таких як AWT, Swing та JavaFX, задовольняє різні рівні складності графічних додатків і активно використовується у додатку Mathpar. AWT, найстаріший інструментарій графічного інтерфейсу, пропонує базові компоненти GUI. Swing, розширення AWT, представляє легкі компоненти, які легко налаштовуються. JavaFX, сучасний стандарт, надає розширені можливості, такі як 3D-графіка та інтеграція мультимедіа, які підходять для сучасних і багатофункціональних інтернет-додатків.

3) Бібліотеки графіки та візуалізації: Окрім стандартних API, Java підтримується потужними сторонніми бібліотеками, такими як JFreeChart або XChart для легкої побудови графіків.

4) Підтримка розробки за допомогою сучасних IDE: Екосистема Java підтримується потужною підтримкою IDE від таких платформ, як Eclipse, IntelliJ IDEA та NetBeans. Ці IDE пропонують складні інструменти для розробки графічного інтерфейсу, включаючи редактори перетягування для Swing і JavaFX, які спрощують процес розробки та допомагають керувати складними проектами.

Таким чином, Java є чудовим вибором для розробки графічних додатків і її використання для імплементації Math Partner дозволяє легко додати необхідний фіункціонал до існуючого застосунку.

2.2. Бібліотека exp4j

При створенні Java-додатків, які вимагають обчислення

математичних виразів та побудови графіків, розробники мають розглянути ряд бібліотек та інструментів, кожна з яких має свої переваги, пристосовані до конкретних потреб. Хоча такі бібліотеки, як JEP (Java Expression Parser), Symja, Apache Commons Math та mXparser мають унікальні можливості, exp4j часто виявляється кращим вибором для певних сценаріїв завдяки своїм специфічним перевагам.

JEP - це комплексна бібліотека для синтаксичного аналізу та обчислення математичних виразів, що підтримує безліч стандартних і користувацьких функцій, змінних і навіть символьне диференціювання. Вона особливо добре підходить для складних математичних обчислень, які виходять за межі базової оцінки. Symja, з іншого боку, пропонує можливості, подібні до тих, що є в Mathematica, включаючи широку підтримку алгебраїчних обчислень, розв'язування рівнянь, що робить її ідеальною для додатків, які потребують надійних алгебраїчних та символьних обчислень.

Apache Commons Math надає набір легких, автономних математичних і статистичних компонентів, які заповнюють прогалини в стандартних математичних функціях Java і Commons Lang, що підходить для широкого спектру додатків, які потребують статистичних і складних математичних обчислень. mXparser - ще один гнучкий варіант з широкою підтримкою синтаксичного аналізу математичних виразів, вбудованих функцій, операторів, констант і функцій, визначених користувачем, що робить його особливо цінним у сценаріях, де складний синтаксичний аналіз і вхідні дані, визначені користувачем, є поширеним явищем.

Бібліотека exp4j - це невелика та ефективна бібліотека Java, яка

чудово справляється з динамічним обчисленням математичних виразів. Вона особливо цінується у програмах, що вимагають розбору та обчислення математичних формул, введених користувачем, що робить її кращим інструментом для розробників, які створюють програми для побудови графіків.

exp4j вміє інтерпретувати математичні вирази, введені у вигляді звичайних рядків. Ця функціональність дозволяє користувачам вводити формулу, наприклад, "sin(x)" або "3 * log(x) - cos(x)", яку exp4j потім обчислює для різних значень x. Ця функція є корисною у програмах побудови графіків, зокрема, тому, що вона підтримує не лише обчислення, але й кастомізацію виразів за допомогою визначених користувачем змінних і користувацьких функцій. Розробники можуть розширювати можливості бібліотеки, додаючи функції, яких не було в exp4j, пристосовуючи її до конкретних вимог програми.

Бібліотека містить повний набір вбудованих функцій, що включає тригонометричні, логарифмічні та експоненціальні функції, які мають суттєве значення для побудови широкого спектру математичних графіків. Розроблений для високої продуктивності, exp4j ідеально підходить для програм, які вимагають обчислень і візуалізації в реальному часі, полегшуючи динамічну побудову графіків функцій та інтерактивну взаємодію з користувачем. exp4j може генерувати необхідні точки даних (x, y) для побудови графіків, які можна легко інтегрувати з графічними бібліотеками, такими як JFreeChart або JavaFX, для візуалізації графіків. Такий модульний підхід спрощує архітектуру програми, роблячи її зрозумілішою та зручнішою для підтримки.

Причини для вибору exp4j серед альтернатив:

- Його легка конструкція та ефективність мають вирішальне значення для додатків у реальному часі або в середовищах з обмеженими ресурсами, що робить його високоефективним для таких завдань, як побудова графіків у веб-застосунку.
- Простий API exp4j дозволяє легко інтегрувати його в додатки, надаючи простий, але потужний інструмент без складнощів, які притаманні більш просунутим бібліотекам, орієнтованим на символьні або алгебраїчні маніпуляції.
- На відміну від бібліотек, які обслуговують ширший спектр математичних операцій, exp4j зосереджується на чисельному оцінюванні, пропонуючи оптимізовану та спрощену функціональність для додатків, які не потребують накладних витрат на символьні обчислення.
- Крім того, легка інтеграція з графічними бібліотеками, такими як JFreeChart або JavaFX, полегшує розробку додатків, які поєднують математичні обчислення з динамічним графічним представленням, покращуючи як функціональність, так і взаємодію з користувачем.
Отже, враховуючи перелічені переваги та особливості застосунку, саме бібліотека exp4j буде використана у цій курсовій роботі.

3. Побудова графіків неявно заданих функцій

3.1 Визначення неявно заданих функцій

Неявно визначені функції - це математичні вирази, в яких змінні не виділені явно з одного боку рівняння. Натомість вони визначаються через рівняння з двома або більше змінними без розв'язання для однієї змінної через інші. Це контрастує з явними функціями, де одна змінна безпосередньо виражається як функція від іншої.

Неявно визначена функція зазвичай представляється у вигляді рівняння виду F(x,y)=0. Тут F - це функція, що включає дві змінні, x та y, і це рівняння стверджує, що зв'язок між x та y неявно визначається парами, для яких F дорівнює нулю.

Приклади:

- Коло: Рівняння кола,

$$x^2 + y^2 = r^2$$

у визначено неявно через x, де r - радіус кола.

- Еліпс: Рівняння

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

визначає еліпс, де a та b - довжини великої та малої осей.

- Гіпербола: рівняння

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$$

описує гіперболу, ще один приклад неявної функції.

Неявні функції можуть описувати складні геометричні фігури та

взаємозв'язки, які важко виразити в явному вигляді. Вони мають вирішальне значення в таких галузях, як фізика, інженерія та економіка, де зв'язки між змінними часто є нелінійними. В обчисленнях неявні функції необхідні для таких інструментів, як неявне диференціювання, яке знаходить похідні, коли розв'язок для однієї змінної є складним.

3.2. Метод "маршируючих квадратів"

У сфері побудови графіків неявно заданих функцій у двовимірному просторі кілька алгоритмів пропонують унікальні переваги залежно від складності функції, необхідної точності та конкретних потреб візуалізації. Існують методи, такі як Ray Casting, контурна побудова, інтервальна арифметика та адаптивне уточнення сітки (AMR), кожен з яких має свої переваги. Ray Casting є високоточним для візуалізації складних структур у швидкозмінних функціях. Контурна побудова, безпосередньо обчислюючи значення на щільній сітці та відстежуючи лінії, де функція задовольняє певну умову, пропонує простоту і легкість реалізації. Інтервальна арифметика забезпечує надійну обробку функцій з важковизначуваними точками, гарантуючи, що жодна деталь не буде пропущена, тоді як AMR фокусується на уточненні роздільної здатності сітки лише там, де це необхідно, підвищуючи ефективність обчислень.

Незважаючи на ці альтернативи, Marching Squares виділяється з кількох причин, особливо при розробці графічних додатків.
Він поєднує в собі ефективність і швидкість, що робить його ідеальним для додатків у режимі реального часу, таких як веб-застосунок. Завдяки використанню таблиці пошуку спрощується розробка, і ця функція підтримується багатьма існуючими бібліотеками та фреймворками, що полегшує подальшу експлуатацію. Marching Squares добре масштабується з різними розмірами координатної сітки, дозволяючи гнучко налаштовувати роздільну здатність відповідно до

обчислювальних ресурсів і бажаного рівня деталізації.

Алгоритм “маршируючих квадратів” є одним з основних методів комп'ютерної графіки, необхідним для вилучення контурних ліній з двовимірних скалярних полів. Цей метод чудово візуалізує неявно задані функції. Метод маршируючих квадратів послідовно обробляє структуровану множину точок даних, визначаючи, де контурні лінії, де (F(x, y) = c), перетинають координатну площину. Цей алгоритм перетворює двовимірне скалярне поле у візуальні контурні лінії.

Ось детальне пояснення того, як працює цей метод:

- Налаштування координатної сітки: Процес починається зі створення скалярного простору, який зазвичай візуалізується у вигляді мережі точок даних, де кожна точка містить певне скалярне значення. Це поле розбивається на сегменти у вигляді звичайної решітки, кожна комірка якої визначається чотирма точками даних у її кутах, що задає базову структуру для контурного мапування.
- Класифікація кутів комірок: Кожен кут комірок сітки класифікується на основі його скалярного значення відносно заданого значення контуру, *c*. Ця класифікація є бінарною; кути позначаються 1, якщо скалярне значення перевищує *c*, і 0, якщо не перевищує, встановлюючи бінарний патерн, який визначає поведінку контурної лінії (графіку функції) всередині комірки.
- Визначення перетинів ребер: Далі алгоритм визначає, які ребра комірки перетинає контурна лінія (графік функції). Це визначення базується на бінарній класифікації: ребро перетинається контуром, якщо його кінцеві точки (кути) мають різну класифікацію (одна більше, а інша менше *c*), що вказує на перетин графіку (лінії контуру).
- Інтерполяція точок перетину: Точки перетину вздовж кожного ребра точно обчислюються за допомогою лінійної інтерполяції.

Наприклад, якщо ребро проходить між двома точками (x_1, y_1) і (x_2, y_2) зі скалярними значеннями v_1 і v_2, точка перетину (x, y) розраховується за формулою:

$$x = x_1 + \frac{c - v_1}{v_2 - v_1} \cdot (x_2 - x_1)$$

$$y = y_1 + \frac{c - v_1}{v_2 - v_1} \cdot (y_2 - y_1)$$

Цей метод обчислює точку вздовж краю комірки, яка пропорційно ділить різницю скалярних значень, точно збігаючись з графіком (лінією контуру; значенням *c*).

- Використання таблиці пошуку: Після визначення ребер, що перетинаються, таблиця пошуку дозволяє визначити, як з'єднати точки перетину в кожній комірці. Ця таблиця відображає 16 можливих бінарних класифікацій кутів клітинок на конкретні шаблони з'єднання між точками, забезпечуючи правильне і послідовне формування сегментів графіку в кожній комірці.

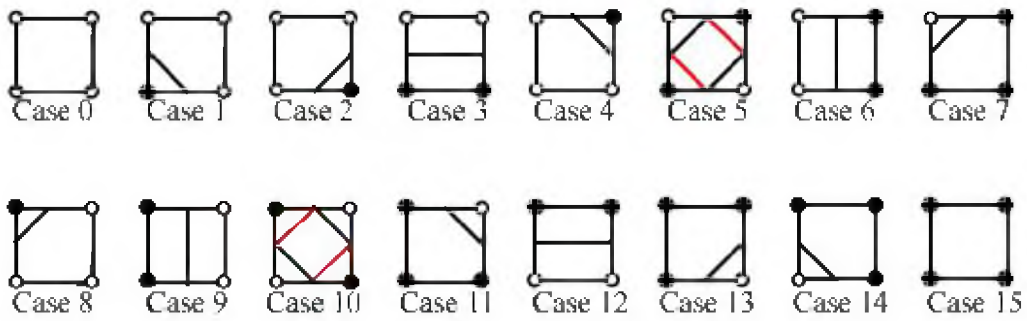

Рис 3.2.1 - Варіації результатів бінарної класифікації комірок для алгоритму Marching Squares

- Побудова контурів графіку: Ці сегменти з'єднуються з сегментами сусідніх комірок, утворюючи безперервні контурні лінії по всій сітці координатної площини. Цей важливий крок гарантує, що візуальне представлення відповідає топології скалярного поля і зберігає геометричну безперервність сітки.
- Обробка особливих випадків: Особливі випадки, такі як комірки, де діагональні кути протилежно класифіковані, можуть призвести

до неоднозначності. Зазвичай ці випадки обробляються шляхом збільшення роздільної здатності сітки (створення більшої кількості комірок меншого розміру - як результат, розбиття "проблемної" комірки на декілька) або застосування спеціальних правил для вибору відповідного шаблону з'єднань, що забезпечує точність і надійність алгоритму.

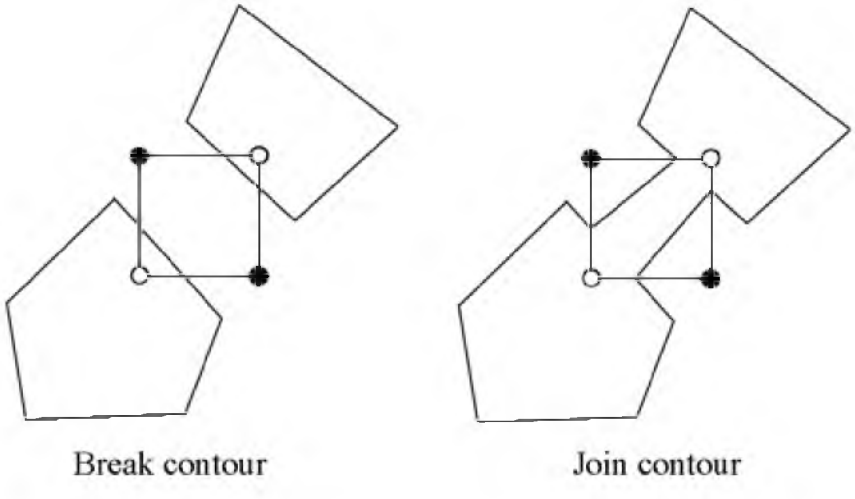

Рис 3.2.2 - Приклад особливого випадку бінарної класифікації комірки для алгоритму Marching Squares

- Фінальний результат: Кінцевим результатом цього процесу є серія контурних ліній, які точно окреслюють заданий рівень контуру *c* по всій координатній площині - тобто шуканий графік функції.

3.3. Імплементація на Java

Для створення взаємодії між браузерною HTML-сторінкою та серверною чатсиною використовується функція draw() визначена у src/main/webapp/js/ plot2d_implicit.js. Взаємодія з користувачем здійснюється за допомогою слухача подій, прив'язаного до кнопки з ідентифікатором submitBtn. При натисканні на цю кнопку функція отримує значення з поля вводу textInput1 і надсилає його у складі POST-

запиту на кінцеву точку сервера за адресою “http://localhost:8080/api/plot2dimplicit”. Очікувана відповідь, масив координат точок, оновлює змінну points, викликаючи функцію drawPoints() для відображення графіку на сторінці.

Java-файл src/main/java/com/mathpar/web/controller/Plots.java містить частину, що описує кінцеву точку REST API, створену за допомогою Spring Framework, яка обробляє POST-запити для побудови 2D-графіків неявних функцій. Метод, анотований за допомогою @RequestMapping(value = "/api/plot2dimplicit", method = RequestMethod.POST), призначає функцію plot2DImplicit як обробник для POST-запитів, спрямованих на URL “/api/plot2dimplicit”. Ця функція повертає об'єкт ResponseEntity, що містить список масивів double[], кожен з об’єктів якого представляє координати точки, разом з відповідним кодом стану HTTP. Функція приймає об'єкт MathparRequest, десеріалізований з тіла запиту. При обробці запиту функція спочатку витягує і записує математичну задачу з об'єкта MathparRequest. Потім вона перевіряє це завдання, щоб переконатися, що воно містить відповідну назву процедури, яка вказує на операцію побудови 2D графіків неявних функцій. Якщо ім'я функції відсутнє, що вказує на неправильно відформатований або неповний запит, метод видає попередження і повертає відповідь “400 Bad Request”. Основна частина функції полягає у виконанні рендерингу, де створюється екземпляр RenderExecutor2d і виконується його метод run з рядком завдання.

RenderExecutor2d у свою чергу передає виконання рендерингу класу Render2d, що інтерпретує математичні вирази через реалізацію інтерфейсу Callable. Таке налаштування уможливлює асинхронне виконання за допомогою ExecutorService, що дозволяє класу повертати результати, не блокуючи основний потік програми.

Основні константи, такі як WIDTH, HEIGHT та SCALE_FACTOR визначають просторові розміри та роздільну здатність, з якою буде обчислюватися функція, впливаючи на щільність сітки комірок. Далі за допомогою звернення до бібліотеки exp4j та методу "маршируючих квадратів" генерується масив з координатами точок, що належать графіку.

Для побудови графіку неявно визначеної функції необхідно використати метод implicitPlot2d.

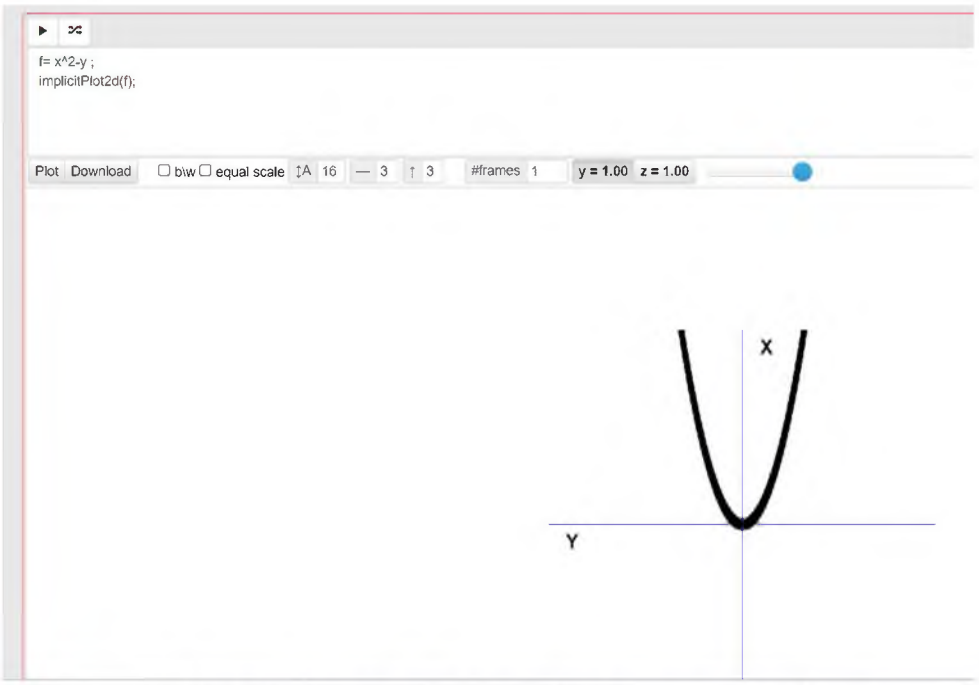

Рис 3.3.1 - Приклад побудови графіку неявно визначеної функції

4. Побудова кількох графіків на малюнку та додавання тексту до графіків

На стороні клієнта визначено JavaScript файл, що відповідає за інтерактивний рендеринг декількох шарів візуаліазції, на кожному з яких знаходиться певний графік. Також у файлі було визначено можливість додавати текст до графіків.

Для візуалізації декількох графіків достатньо визначити декілька функцій f і кожна з них буде послідовно оброблена та додана до координатної сітки.

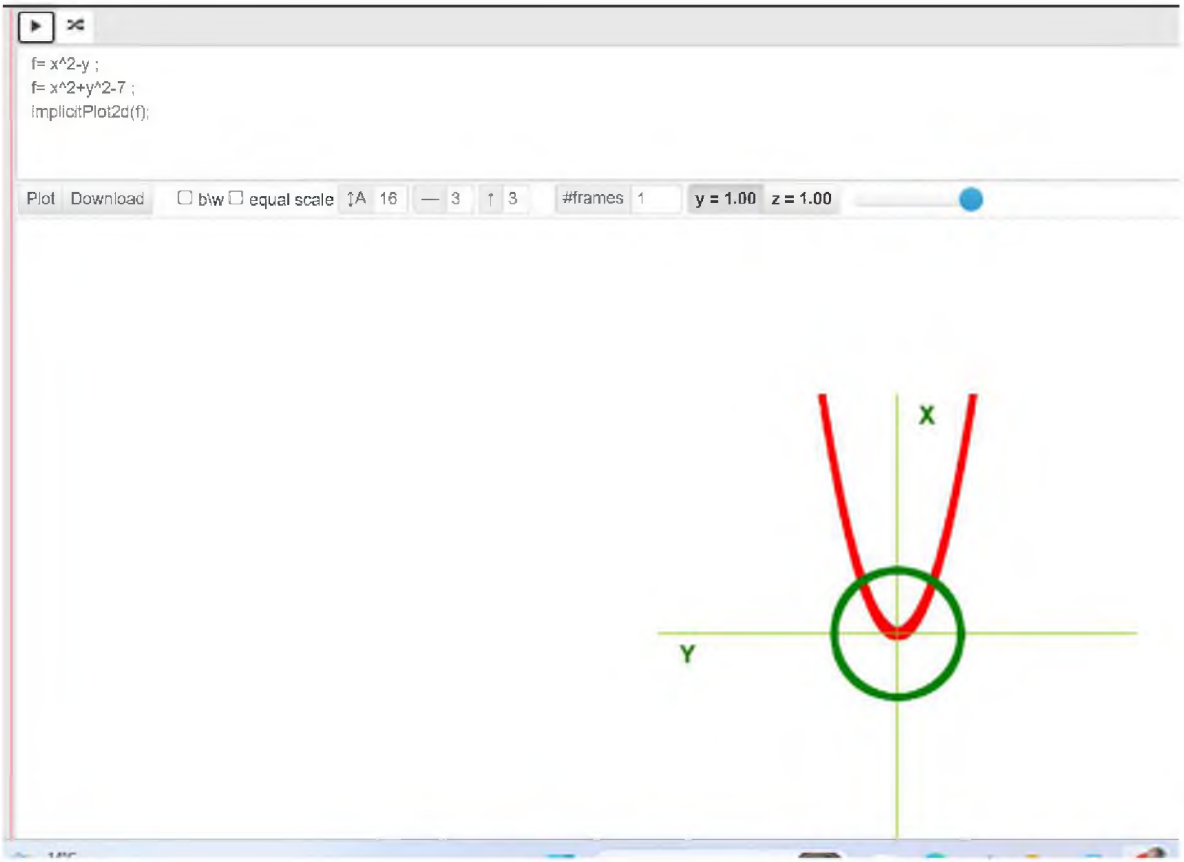

Рис 4.1. - Приклад побудови декількох графіків на малюнку

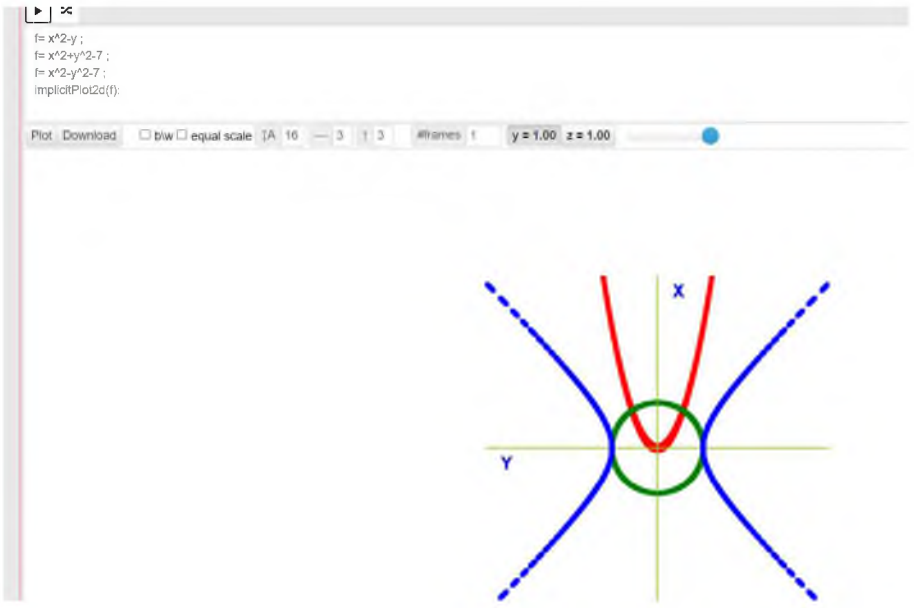

Рис 4.2. - Приклад побудови декількох графіків на малюнку

Для додавання тексту треба визначити елементи title (= “chart1”) або text (=”text1”).

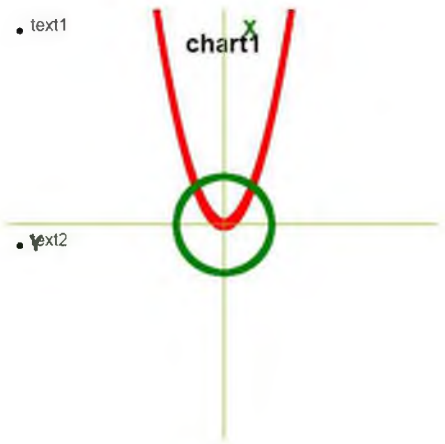

Рис 4.3. - Приклад додавання тексту до графіку

Висновки

Результатом виконання цієї роботи стало розширення фукціоналу додатку Math Partner можливостями побудови неявно визначених функцій, створенню декількох графіків на одній координатній площині, а також додавання текстових приміток до графіку.

Цей функціонал є особливо корисним для користувачів в освітній та професійній сферах. Наприклад, викладачі можуть використовувати ці вдосконалені інструменти для візуальної

демонстрації понять з математики чи фізики, сприяючи глибшому розумінню студентами.

Під час розробки цього додатку було досліджено та використано бібліотеки мови Java, такі як exp4j та JavaFX, які надають необхідні обчислювальні та графічні можливості для реалізації описаних вище розширених функцій. Проект включав поглиблене вивчення чисельних методів і графічних алгоритмів, необхідних для точного рендерингу неявно визначених функцій, фокусуючись на методі “маршируючих квадратів”.

Крім того, нові можливості були інтегровані із існуючим інтерфейсом користувача, дозволяючи взаємодіяти із графіками більш ефективно.

Ця програма має потенціал для подальшого розвитку. Найближчі перспективи включають вдосконалення графічного інтерфейсу та розширення можливостей налаштування текстових анотацій.

На закінчення, ця курсова робота не тільки демонструє можливості додатку в покращенні візуалізації та взаємодії зі складними математичними функціями, але й підкреслює гнучкість та потужність Java в розробці та підтримці складних освітніх та аналітичних інструментів.

Список використаної літератури

1) Impact of graph technologies in K-12 science and mathematics education [електронний ресурс] https://www.sciencedirect.com/science/article/abs/pii/S036013151930301X
2) Marching Squares Demonstration [електронний ресурс] https://urbanspr1nter.github.io/marchingsquares/#:~:text=URL%3A%20https%3A%2F%2Furbanspr1nter,100
3) Офіційна документація бібліотеки exp4j [електронний ресурс] https://www.objecthunter.net/exp4j/
4) Хмарна математика MathPartner у Києво-Могилянській академії / Г. І. Малашонок // Наукові записки НаУКМА. Комп'ютерні науки. - 2017. - Т. 198. - С. 27-35. - Посилання: http://nbuv.gov.ua/UJRN/NaUKMAkn_2017_198_8
5) Архітектура Math Partner [електронний ресурс] Г. І. Малашонок

(2) Architecture_Cloud_Math 2023 - YouTube