

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

Тимошук Володимир Петрович

**Програмний модуль візуалізації графіків
математичних функцій довільної складності / The software
module for graphs visualization of mathematical functions by
various complexity**

напрямок підготовки: 6.050102 - Комп'ютерна інженерія
фахове спрямування - Комп'ютерні системи та мережі
Бакалаврська робота

Виконав студент групи КСМз-41/2
Тимошук Володимир Петрович

Науковий керівник:
Батько Ю.М.

Тернопіль - 2018

РЕЗЮМЕ

Дипломний проект містить 58 сторінок пояснюючої записки, 11 рисунків, 12 таблиць, 2 додатки. Обсяг графічного матеріалу 2 аркуші формату А3.

Метою дипломної роботи є розробка програмного модуля візуалізації графіків математичних функцій довільної складності.

Методи досліджень базуються на теорії алгоритмів (для аналізу розроблених методів та алгоритмів), алгоритмах сегментації (для виділення окремих символів ввхідній стрічці), технологій об'єктно-орієнтованого програмування (для програмної реалізації розробленого програмного модуля).

В дипломній роботі на основі аналізу існуючих алгоритмів структурного аналізу та алгоритмів розпізнавання математичних виразів було спроектовано та реалізовано програмний модуль візуалізації графіків математичних функцій. В якості вхідних даних було обрано математичні функції довільної складності.

Проведено тестування розробленого програмного додатку на мові програмування Delphi, що підтвердило доцільність використання структурного підходу при проектуванні та реалізації програмних систем обробки та аналізу математичних виразів.

Розроблений програмний продукт є ефективним засобом з простим інтерфейсом, що дозволяє вирішувати проблему обробки та візуалізації математичних формул та може бути використаний при побудові програмних систем.

Ключові слова: ГРАФІК ФУНКЦІЇ, СЕМАНТИЧНИЙ АНАЛІЗ, РОЗПІЗНАВАННЯ ФОРМУЛ, ВІЗУАЛІЗАЦІЯ ДАНИХ.

RESUME

Diploma project contains 58 pages of explanatory notes, 11 figures, 12 table 2 applications. Volume of graphic material 2 leaves of format A3.

The meta of the thesis is to develop a software module for visualization graphs of mathematical functions of arbitrary complexity.

Research methods are based on the theory of algorithms (for the analysis of developed methods and algorithms), segmentation algorithms (for the selection of separate characters in the input tape), technology object-oriented programming (for software implementation of the developed software module).

In the thesis work on the basis of analysis of existing algorithms of structural analysis and algorithms for mathematical recognition, a software module for rendering graphs of mathematical functions was designed and implemented. As input data, mathematical functions of arbitrary complexity were selected.

The testing of the developed software application in the programming language Delphi has been tested, which confirmed the expediency of using the structural approach in the design and implementation of software systems for processing and analysis of mathematical expressions.

The developed software product is an effective tool with a simple interface that allows you to solve the problem of processing and visualization of mathematical formulas and can be used in the construction of software systems.

Keywords: GRAPHIC FUNCTIONS, SEMANTIC ANALYSIS, FUNCTIONS RECOGNITION, DATA VISUALIZATION.

ЗМІСТ

Перелік умовних скорочень	9
Вступ.....	10
1 Технології та програмні засоби обробки математичних виразів	12
1.1 Аналіз підходів задання математичних формул	12
1.2 Класифікація основних типів графіків.....	14
1.3 Аналіз програмних засобів розпізнавання математичних виразів.....	19
1.4 Висновки та постановка задачі	22
2 Проектування програмного модуля візуалізації графіків складних математичних виразів	23
2.1 Алгоритми структурного аналізу для математичних формул.....	23
2.2 Алгоритм аналізу математичних формул.....	26
2.3 Моделювання програмного модуля візуалізації графіків.....	32
3 Програмна реалізація програмного модуля візуалізації графіків математичних виразів.....	35
3.1 Опис структури програмного додатку	35
3.2 Реалізація методів розпізнавання та візуалізації математичних виразів ..	40
3.3 Тестування програмного модуля візуалізації складних графіків.....	43
4 Техніко-економічний розділ	46
4.1 Розрахунок витрат на розробку програмного додатку.....	46
4.2 Визначення експлуатаційних витрат... Помилка! Закладку не визначено.	
4.3 Визначення економічної ефективності і терміну окупності капітальних вкладень..... Помилка! Закладку не визначено.	
Висновки	57
Список використаних джерел	58
Додаток А	Помилка! Закладку не визначено.
Додаток Б.....	Помилка! Закладку не визначено.

					ДР.КСМ.110850/16.00.00.000 ПЗ		
Змн.	Лист	№ докум.	Підпис	Дата			
Розробив		Гимошук В.П.			Літ.	Арк.	Акрушів
Перевір.		Батько Ю.М.			8	56	
Консульт.		Паздрій І.Р.			ТНЕУ, ФКІТ, КСМЗ-41/2		
Н. Контр.		Гураль І.В.					
Затвердив		Березький О.М.					

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ОС	–	Операційна система
БД	–	База даних
ПЗ	–	Програмне забезпечення
UML	–	Unified Modeling Language

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Візуалізація даних - це наочне уявлення масивів різної інформації, причому існує кілька типів візуалізації. Звичайне візуальне уявлення кількісної інформації в схематичне формі. До цієї групи можна віднести всім відомі кругові та лінійні діаграми, гістограми і спектрограми, таблиці і різні точкові графіки. Дані при візуалізації можуть бути перетворені в форму, посилює сприйняття і аналіз цієї інформації. Наприклад, карта і полярний графік, тимчасова лінія і графік з паралельними осями, діаграма Ейлера. Концептуальна візуалізація дозволяє розробляти складні концепції, ідеї і плани за допомогою концептуальних карт, діаграм Ганта, графів з мінімальним шляхом та інших подібних видів діаграм. Стратегічна візуалізація переводить в візуальну форму різні дані про аспекти роботи організацій. Це всілякі діаграми продуктивності, життєвого циклу і графіки структур організацій. Графічно організувати структурну інформацію за допомогою пірамід, дерев і карт даних допоможе метафорична візуалізація, яскравим прикладом якої є карта метро. Комбінована візуалізація дозволяє об'єднати кілька складних графіків в одну схему, як в карті з прогнозом погоди.

Однією з відмінних рис текстового документа від математичного є наявність в останньому відповідного набору символів: знаків математичних операцій, букв грецького алфавіту, спеціальних математичних символів (знаки інтеграла, дужки, знаки операцій). Також для математичної формули характерно використання символів різного розміру (індекси та ступеня). Деякі формули можуть мати в складі символи, розташовані на декількох рядках або по довжині перевищують середню довжину символу або пробілу.

Візуальна інформація краще сприймається та дозволяє швидко і ефективно донести до глядача власні думки та ідеї. Фізіологічно, сприйняття візуальної інформації є основною для людини. Є численні дослідження, які підтверджують, що:

- 90% інформації людина сприймає через зір;

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

- 70% сенсорних рецепторів знаходяться в очах;
 - близько половини нейронів головного мозку людини задіяні в обробці візуальної інформації;
 - на 19% менше при роботі з візуальними даними використовується когнітивна функція мозку, що відповідає за обробку та аналіз інформації;
 - на 17% вище продуктивність людини, що працює з візуальною інформацією;
 - на 4,5% краще згадуються докладні деталі візуальної інформації;
- Тому задача створення програмного додатку розпізнавання математичних виразів та побудови графіків функцій є актуальною.

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

1 ТЕХНОЛОГІЇ ТА ПРОГРАМНІ ЗАСОБИ ОБРОБКИ МАТЕМАТИЧНИХ ВИРАЗІВ

1.1 Аналіз підходів задання математичних формул

Розпізнавання формули дає різні можливості для користувачів, серед яких:

- пошук наукових документів по математичному описі, формулах та підформулах;
- формування бази знань формул;
- переміщення формул в зовнішні документи;
- отримання семантичної інформації з формул;
- інтеграція з системами комп'ютерної алгебри тощо.

В даний час не існує повномасштабного проекту аналізу математичних формул, який би був безкоштовним для використання та мав відкритий вихідний код. Сучасні проекти є скоріш не окремими програмними продуктами, а інструментами оптичного розпізнавання, що використовуються сторонніми системами. Використання контенту документа додаткових інформаційних характеристик самого документа (метаданих) призначено для зменшення залежності від складної та неточної процедури оптичного розпізнавання тексту.

Визначити формулу в документі означає задати межу області, що містить всі елементи формули, розпізнати символи, що входять в дану область, згрупувати символи в змістові одиниці, відновити ієрархічну структуру формули. Отримана структура може бути переконвертована в будь-який формат запису формули.

Математичні вирази представляють собою двомірну ієрархічну структуру даних. Формули можуть містити:

- символи латинського алфавіту;
- символи грецького алфавіту;
- спеціальні символи;

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

- графічні об'єкти.

Відзначимо, що зображення формули не формалізовано строго, у представленні формули відсутня канонічна форма запису. Точне визначення математичного виразу вимагає задання формального списку математичних символів в базі знань формул.

Як правило, в документах, що містять формули вилучаються метадані: об'єкти, зображення або набори графічних команд, що використовуються для візуалізації символів. Метадані можуть виступати як елементи множини шрифтів, так і об'єкти зображення. Між метаданими та символами не існує однозначної відповідності: символ може бути описаний одним або декількома метаданими. Значення простих метаданих визначається за таблицею символів або за допомогою геометричної інтерпретації графічних об'єктів. Композитні метадані визначаються евристичними правилами оптичного розпізнавання.

При аналізі, найчастіше використовується геометрична сегментація формула на початковій стадії аналізу для декомпозиції елементів сторінки, наприклад, метод розбивання на профіль проекції [6] (різання проекційного профілю). Однак геометрична сегментація часто не є достатньою: потрібно додатковий аналіз для уточнення сегментації.

Складність визначення елементів полягає у використанні формульних змінних. Зв'язки між токенами задаються операторами. В математичній нотації існують неявні оператори, задані лише відносним розташуванням частин формул, розпізнавання яких ускладнюється відсутністю стандарту зображення неявних операторів.

Однак існують властивості математичних записів, які допомагають ідентифікувати та розпізнати формули. Ці геометричні властивості (відступи, вирівнювання), властивості написання (використання математичних шрифтів, курсив або жирне написання), контекст.

Ці властивості дозволяють використовувати гербові правила аналізу. До розпізнавання формул адаптуються статистичні методи розпізнавання: за навчальними копіями формул [2], гістограма визначених параметрів (наприклад, пропускових відстаней) [1], специфічні властивості можуть бути

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

використані як вектори в методі опорних векторів [4], застосовуються для алгоритмів навчання. Методи можуть розглядатися в комбінаціях один з одним. Для розпізнавання відносин між ним використовуються різноманітні розширення граматики (координатна граматика Андерсона, двомірні граматики, графові граматики).

1.2 Класифікація основних типів графіків

Трактовка графічного методу як особливої знакової системи – це, штучна знакова мова - пов'язана з розвитком семіотики, науки про знаки та знакові системи, що використовуються для передачі даних між адресатами. Знак в семіотиці служить символічним вираженням деяких явищ, властивостей або відносин. Існуючі в семіотиці знакові системи прийнято розділяти на немовні та мовні. Немовні знакові системи - це специфічні системи, які складаються з певних символів, умовних знаків, створених людиною в процесі пізнання навколишньої дійсності (наприклад, шкала вимірювального приладу, висота стовпця ртуті в термометрі тощо). Характерним для цих знакових систем є те, що поєднання знаків в них набуває сенсу лише тоді, коли вони об'єднуються за певними правилами. Ці знакові системи не мають загальноприйнятого характеру, а набувають практичних значень в рамках деяких областей знань.

Мовні знакові системи, або мови, є найбільш важливими та складнішими знаковими системами передачі інформації, ніж немовні знакові системи. Мовні поділяються на природні (жива людська мова та знакові системи тварин) і штучні. З точки зору семіотики, людська мова, виражена знаками-буквами, складається з природньої мови, штучні мовні системи використовуються в різних областях життєдіяльності та техніки. До таким мов відносяться:

- графіки;
- системи математичних формул;
- хімічних знаків;

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

- алгоритмічні мови.

Штучні або символні мови разом з природньою мовою, спрощують запис спеціальних питань певної області знань.

Графіки знайшли широке застосування для передачі різноманітних родів інформації в різних сферах життя, що зумовило різноманітність їх видів. Це привело до багатозначності поняття графіка, відсутності в даний час чітких критеріїв термінології та класифікації.

Графічний образ (основа графіків) - це геометричні знаки, тобто сукупність точок, ліній, фігур, з допомогою яких відображені статистичні показники. Важливо правильно вибрати графічний образ, який повинен відповідати графічним цілям і сприяти найбільшій виразності зображення. Графічні - це ті образи, в яких властивості геометричних знаків - фігура, розмір ліній, розташування частин - мають істотне значення для вираження вмісту відображених величин, причому кожному зміненню вираженого змісту відповідає зміна графічного образу.

Поле графіків - це частина площини, де розташовані графічні образи. Поле графіки має певні розміри, які залежать від його призначення. Рекомендується використовувати графіку поля з співвідношенням короткої та довгої сторони 1:1,41. Таке співвідношення прийнято для Міжнародної організації з стандартизації (ISO).

Просторові орієнтири графіка задаються у вигляді системи координатних сіток. Система координат необхідна для розміщення геометричних знаків у графічному полі. Система координат - це сукупність елементів, що визначають положення точки на прямій або кривій лінії, на площині або в просторі. Існують різні системи координат. Найпоширенішою є система прямокутних (декартових) координат через простоту її побудови, виразності різних співвідношень і залежностей між зображеннями величин. Прямокутна система координат утворюється сукупністю двох пересічних перпендикулярних площин, що називаються осями координат.

Полярная система координат будується визначеної точки 0, що називається полюсом або центром обертання. Для зручності побудови та

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

читання діаграм у полярній системі координат побудовано координатну сітку у вигляді концентричних округів з центром у полюсі.

Масштаб графіка - це міра переведення числової величини в графічну. Масштабною шкалою називається лінія, окремі точки якої можуть бути прочитані як певні числа. Шкала має велике значення в графіку і включає три елементи: лінія - носій шкали; визначений ряд графічних інтервалів, розташованих на носії шкал в певному порядку; цифрове позначення чисел, що відповідає графічним інтервалам.

Одним з видів графіків є графік наочного зображення кількісної залежності різних масових явищ, процесів тощо. Назви цих графіків різноманітні: статистичні графіки, діаграми, статистичні діаграми.

Другим видом графіків є номограми - розрахункові графіки, метою яких є обчислення результатів при будь-яких комбінаціях наданих значень змінних, від яких цей результат залежить. Номограми є зручним обчислювальним інструментом.

До інших видів графіків відносяться органи - структурні схеми організації підприємства; графіки руху транспорту; графіки-розкладу роботи підприємств, контрольно-планові графіки організації виробництва тощо.

Діаграма представляє собою малюнок, на якому сукупності, що характеризуються певними показниками, з метою їх узагальнення та аналізу описуються за допомогою умовних геометричних образів або знаків, що є графічною мовою.

Графічна мова має свої специфічні особливості, які відрізняють її від інших штучних мов. До такої особливості відноситься двовимірність запису: при передачі інформації графічною мовою використовуються два виміри - лінійна послідовність розміщення знаків (стрічка або ряд стрічок) та їх взаємозв'язок на площині.

Іншою особливістю графічного слова є безперервність вираження, що проявляється в тому, що інформація, передана за допомогою графічного мови, представляється за допомогою системи взаємозв'язаних знаків, а не окремих лінійно розташованих дискретних знаків. Мова графіків істотно відрізняється

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

від інших штучних мов, наприклад, математичного, для якого характерна дискретність знаків і лінійна послідовність їх розташування.

Ще однією особливістю графічного слова є окремість викладу. Інформація, передана в графічному вигляді, як правило, відособлено від безпосередньо пов'язаної з нею за змістом інформації, представленої у словесній або письмовій формі - текстовою формою. Об'єктивність викладу графічного образу безпосередньо впливає з його природи, його здатності адекватно передавати кількісні та відособлені якісні характеристики досліджуваних явищ та виявляти з вихідної інформації нові властивості та особливості, що знаходяться в ній у прихованому вигляді.

До особливостей графічної мови також відноситься її метричність та наочність. Метричність - використання в графіках масштабних шкал та умовних позначень, дозволяє визначити окремі показники, рівні та розміри досліджуваних явищ. Представлення інформації у вигляді графіків більш наочно та доступно, ніж табличне, це дозволяє краще осмислити результати спостереження, правильно їх інтерпретувати, отримати нові знання про предмет дослідження, загальну вихідну інформацію.

Суттєвою особливістю графічного тексту є його сумісність - легкість та гнучкість об'єднання знаків графічної мови між собою та з знаками інших мовних та немовних знакових систем.

Форми графічного різноманітного образу: геометричні та фігурні (негеометричні) знаки з плоскими або об'ємним зображенням. У відповідності з цим розрізняють графи точкові, лінійні, плоскі та просторові (об'ємні).

При побудові точних діаграм в якості графічних образів застосовуються сукупності точок; при побудові лінійних - ліній, ізоліній. Основний принцип побудови всіх плоских діаграм зводиться до того, що величини зображуються у вигляді геометричних фігур, що підрозділяються на стовбці, полоски, кола, сектори, квадрати та фігури. Ці же принципи побудови відносяться і до об'ємних графіків, крім того, до них відноситься особлива форма об'ємного графічного образу - поверхневий розподіл, що відображає залежність одночасно трьох величин.

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

Класифікація графіків за формою графічного образу наведена на рисунку

1.1.

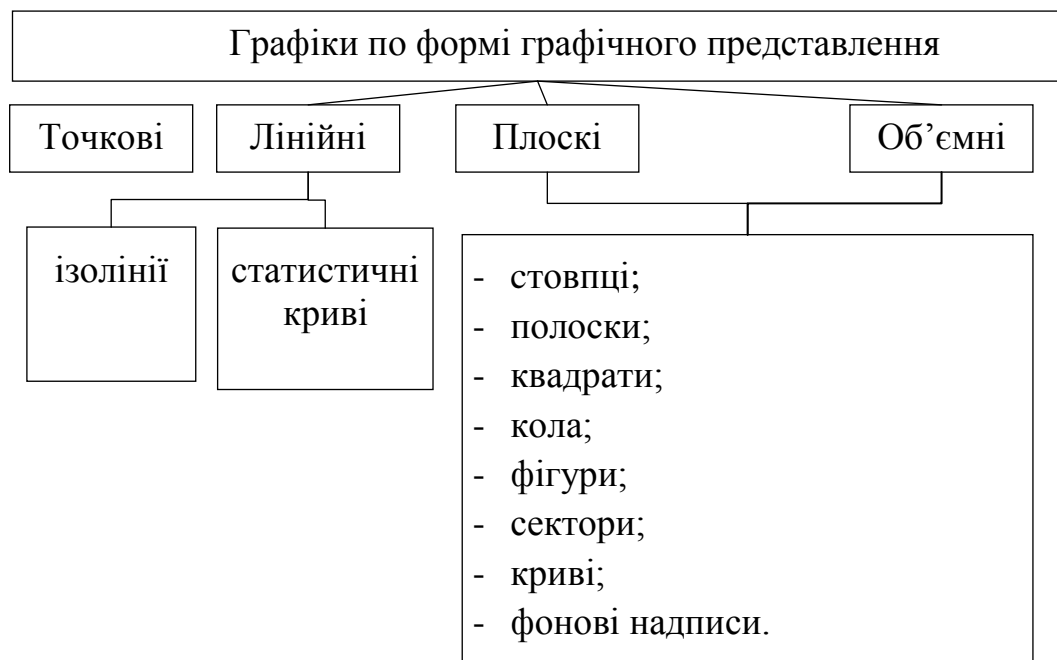


Рисунок 1.1 - Класифікація графіків за формою графічного образу

По способу побудови та завдань, для яких використовуються графіки поділяються на діаграми, графічні карти, контрольні карти, взаємозв'язані графіки (рисунок 1.2).

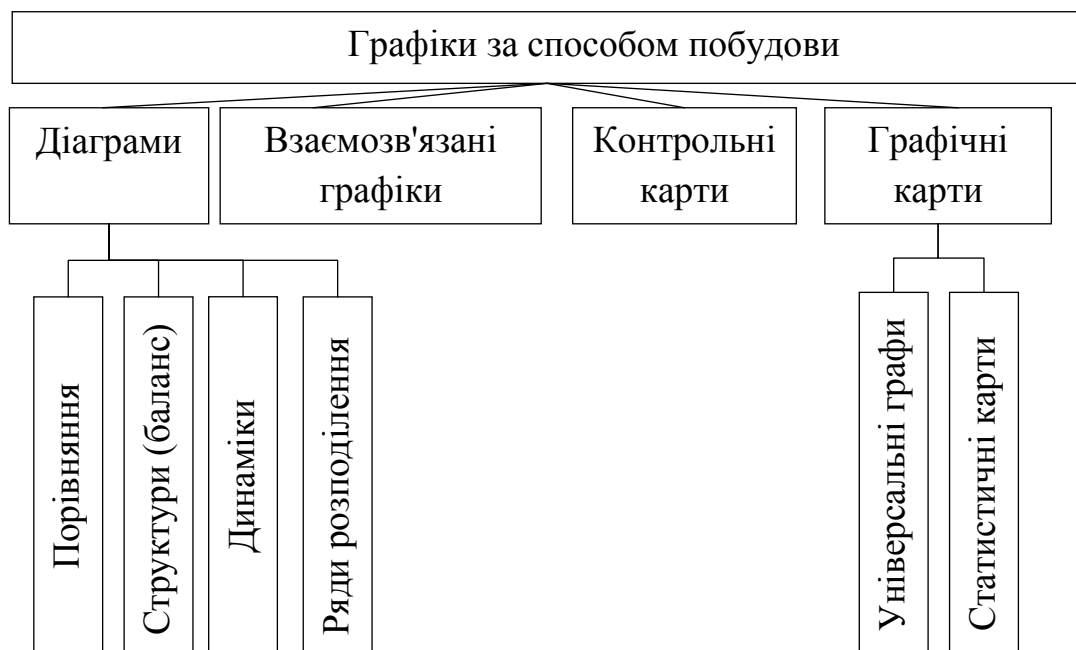


Рисунок 1.2 - Класифікація графіків по способу побудови та задачам

Діаграми є найбільш поширеним способом графічних зображень. Вони - графіки кількісних відносин. Діаграми застосовуються для налагодженого співставлення в різних аспектах (просторовому, тимчасовому тощо) незалежних один від одного величин: виготовлення електроенергії, генерації потужності, споживаної електроенергії тощо. При цьому порівняння досліджуваних сукупностей здійснюється за будь-якою істотною змінною ознакою.

В залежності від кола вирішуваних задач виділяють діаграми порівняння, структури, динаміки, накопичення, ряди розподілів величин варіаційного ряду.

Графічні карти - графіки кількісного розподілу ознак на площині або в часі. По своїй основній цілі вони подібні до діаграм і специфічні тільки в тому відношенні, що представляють собою умовні зображення даних на карті простору або часу, тобто відображають просторове або часове розміщення даних. Вони підрозділяються на універсальні графіки та статистичні карти (картографи, картодіаграми).

Контрольні карти - це вид графічних зображень та контрольного діапазону, які дозволяють проводити поточний контроль виробничого процесу на промисловому підприємстві та прогнозувати його розвиток.

Взаємозв'язні графіки - вид графічних зображень, що відображають тимчасові зміни взаємозв'язків різноманітних показників. Їх поділяють на накопичувальні та технологічні. Перераховані види графіків не є вичерпними, але вони найчастіше використовуються.

1.3 Аналіз програмних засобів розпізнавання математичних виразів

Програми, призначені для підготовки наукових публікацій з використанням формул та спеціальної символіки, представляють традиційний інтерес для багатьох категорій користувачів. OCR-системи давно стали поширеними програмними продуктами на ринку програмного забезпечення та

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

успішно використовуються для обробки текстів, анкет, табличних даних, штрихкодів тощо. Розпізнавання математичної символіки до цих пір є в цілому невіршеним питанням через складну структуру формул і, відповідно, необхідності застосування дуже гнучких алгоритмів їх інтерпретації.

Програма InftyEditor (рисунок 1.3) це достатньо багатофункціональний текстовий редактор з підтримкою трьох режимів для введення звичайного тексту, математичних формул і традиційних команд TeX. Документи зберігаються у форматах HTML, MathML та спеціалізованому Karlsruhe AMS, а також у власному * .iml-форматі на базі XML, за винятком формул, у них також допускається вставити зображення в форматах BMP, TIFF, JPG, PNG. Зображення традиційно вводиться або безпосередньо з сканера, либо з графічного файлу в форматі TIFF або PNG з розширенням 300-600 точок на дюйм.

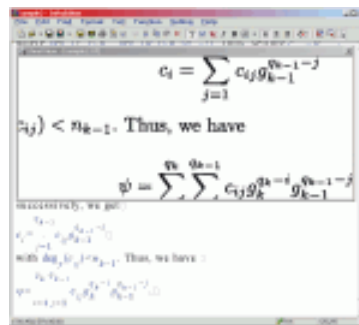


Рисунок 1.3 – Робоче вікно програми InftyEditor

Подальший процес розпізнавання складається з чотирьох фаз: аналізу сторінки макета, розпізнавання формули та тексту, структурного аналізу математичних виразів та ручної корекції. Аналіз макета полягає у перекодуванні зображення у внутрішній цифровий формат, очищення його від «цифрового сміття» та виявлення складових елементів сторінки - таблиць, рисунків, текстових блоків. На етапі розпізнавання відбувається відділення тексту від математичних формул. Структурна фаза зводиться до аналізу математичних формул та їх представлення в деякому внутрішньому форматі, придатному для відображення на екрані та експорту у зовнішній файл. Ручна корекція дозволяє виправляти помилки розпізнавання та додавати нові дані.

Середній відсоток помилок (в тексті та формулах) не перевищує 1-2%, а швидкість опрацювання зображень на сучасних комп'ютерах – до декількох сторінок за хвилину.

MatanCrack. Програма для розпізнавання спеціального зображення у вигляді математичної формули. В якості математичної формули використовується інтеграл. Програма показує хороший результат роботи: 93% правильних відповідей при вибірці з 500 штук.

Недоліки програми:

- опрацюються тільки чорні символи на білому фоні;
- необхідна відсутність шуму та інших артефактів (наприклад, ліній);
- необхідно, щоб символи ніколи не перетинались;
- необхідні однакові шрифти;
- степені та множники складаються з одного цифри;
- степені і множники знаходяться в діапазоні від 2 до 5.

Інтернет-сервіс anti-captcha.com надає можливість розпізнавання спеціального зображення, використовує програмний інтерфейс (API) системи. Система розпізнає велику кількість різних спеціальних образів. Також її перевагою є те, що всі розрахунки відбуваються на стороні сервісу. Таким чином, трапляються ресурси клієнта. Але недоліком є закритість системи. Так само система є комерційною, а її використання - платне.

AddSite. Потужна система для розпізнавання образів. Підтримує велику кількість різних методів. Програма має велику кількість закладених алгоритмів, які справляються зі складними зображеннями. Недоліком, так само як і у багатьох інших рішень, є її ціна. Програма комерційна та має закритий код.

PWNtcha. Відмінна програма з відкритим вихідним кодом. Має хорошу алгоритмічну базу. Розпізнає велику кількість типових зображень. Проте програма більше не підтримується розробниками.

aiCaptcha. Хороша програма для розпізнавання. Перевагою даного програмного продукту, що вона вміє розпізнавати зображення з динамічним фоном. Символи можуть бути зміщені по своїй осі на довільну кількість градусів. Недоліком є її закритість та масштабованість. Також, даний

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

програмний продукт, здатний розпізнавати лише вузькоспеціалізовані зображення.

В результаті проведеного аналізу визначено такі основні характеристики програмних додатків розпізнавання математичних виразів: наявність функції розпізнавання виразів різної структури, перекодування формул в внутрішній формат, можливість ручної корекції отриманого результату.

1.4 Висновки та постановка задачі

В даному розділі проаналізовано задачі, що ставляться перед розробниками програмних засобів візуалізації графіків на основі автоматичного розпізнавання математичних виразів. Проведено дослідження та класифікацію існуючих підходів до задання графіків за допомогою математичних записів. Проведено аналіз та класифікацію існуючих видів графіків та визначено основні вимоги до промальовки графіків за допомогою цифрової техніки. Розглянуто сучасні програмні системи для аналізу математичних виразів, виділено їх основні переваги та недоліки.

Для досягнення поставленої мети необхідно розв'язати наступні задачі:

1. Провести аналіз підходів до задання математичних виразів.
2. Провести класифікацію існуючих видів графіків.
3. Проаналізувати існуючі програмні засоби розпізнавання та візуалізації математичних формул.
4. Розробити алгоритм аналізу математичних виразів.
5. Спроекувати структуру програмного модуль візуалізації математичних формул довільної складності.
6. Реалізувати програмний модуль візуалізації математичних формул, провести його тестування й порівняти з програмами аналогами.

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

2 ПРОЕКТУВАННЯ ПРОГРАМНОГО МОДУЛЯ ВІЗУАЛІЗАЦІЇ ГРАФІКІВ СКЛАДНИХ МАТЕМАТИЧНИХ ВИРАЗІВ

2.1 Алгоритми структурного аналізу для математичних формул

Формулою будемо називати вираз, що має наступний вигляд:

<ліва частина> <оператор> <права частина> ,

де <оператор> - це арифметична або логічна операція або знак відносин;

<ліва частина> і <права частина> - вирази (підформули).

В свою чергу підформули, як і формула, можуть мати такийже вид.

Формула складається з символів, до яких відносяться цифри, букви латинського та грецького алфавітів, знаки арифметичних та логічних операцій, стрілки, скобки, спеціальні знаки, що використовуються в математичних формулах (оператори з змінним діапазоном).

По своєму розташуванню в тексті формули поділяються на два типи: однострокові, вбудовані в нематематичний текст (наприклад, «... рівняння прямої має вид $y = kx + b$, де ...»), і багатострокові, що знаходяться в ізольованій частині між частинами нематематичного тексту, наприклад:

$$y(x) = \frac{e^x - e^{-x}}{4}$$

На першому етапом аналізу формули є її розбиття на окремі символи. Процес отримання на виході зображень із окремим символом формули складається з наступних кроків:

- читання (ReadImages) вихідного зображення з пам'яті;
- видалення фону (HistDelete) на вхідному кольоровому зображенні шляхом аналізу гістограми. Отримане кольорове зображення, в якому фон представлений альфа-каналом, подає на сегментацію;

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

- сегментація (FindObjects, GetObjects) базується на позначенні зв'язних компонент на кольоровому зображенні. Цей метод ефективно працює на символах, що складаються з однієї компоненти (наприклад, 1, 2, A, B), а для символів з декількох компонентів (j, i, =, %) потрібна додаткова обробка для об'єднання, перш ніж символи будуть подані на розпізнавання;

- передача (ImageWriter) отриманих сегментів формул на наступний етап обробки, що використовує на вході серію образів.

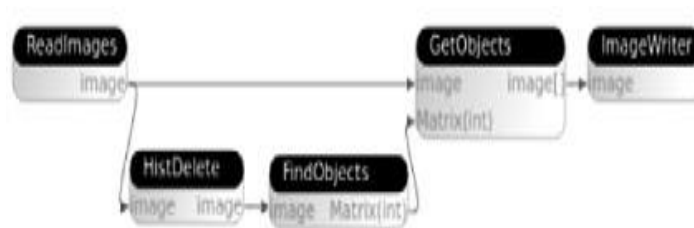


Рисунок 2.1 - Схема сегментації окремих елементів формули

Для опису символів, представлених матрицею точок, виконується процедура скелетизації. Скелет представляє собою плоский граф, що відбиває особливості форми об'єкта [12]. Вершини скелета - це центри окружності, що стосуються границь фігур в трьох або більше точках, а ребра - лінії, що складаються з точок-центрів вписаних округів, що відносяться до двох або більше точок. Для виявлення скелетів попередньо бінарізованого після сегментації зображення використано алгоритм Зонга-Суня [14]. При скануванні символу застосовується 8-зв'язна структура сусідства пікселей. Перефарбовуються в білий колір ті чорні граничні точки, які не порушують зв'язків залишків фігури. Алгоритм скелетизації спрощує подальшу роботу з зображенням і стійкий до шуму. До недоліків даного перетворення символів відносяться вірогідність порушень зв'язку та відносно великих тимчасових витрат.

Наступний крок полягає в пошуку ліній у скелетному зображенні для вилучення інформації про вміст у формулі символів дробів, негативних значень, знаків «=», «-». Якщо такі дані присутні, то проводиться дослідження навколишніх символів.

Для виділення лінії на зображення використовується перетворення Хафі [8]. Прямі описуються за допомогою рівнянь $r=x\cos\theta+y\sin\theta$ при умові, що $\theta\in[0,\pi]$ і $r\in\mathbb{R}$, де r - величина вектора від початкового координат до найближчого крапку на прямій, а θ - кут між цим вектором і вісю координат.

Наступним кроком структурного аналізу формули є розбивання її на підформули. Розбиття формул відбувається до тих пір, поки не залишається не ділимого на символи виразу. Кожен символ, що з'єднує дві підформули, має свій пріоритет, відповідно до якого визначається послідовність пошуку символу в формулі (пріоритет "1" – для "=", пріоритет «2» - для стрілки, пріоритет «3» - для знаків "більше" або "менше", пріоритет «4» - для знаків арифметичних та логічних операцій). Додатковими символами підключення підформули є: оператори з змінним діапазоном, знак корня, горизонтальна риска та дужки.

За допомогою методу базових локальних ознак [8] формується множина ознак розпізнаваних об'єктів, характерних для певного символу. В якості множини ознак будуть виступати кутові точки, знайдені на символі. Розпізнавання об'єкта відбувається шляхом зіставлення ознак моделі та ознак зображення.

Семантичною мережею (деревом) назвемо інформаційну модель предметної області, представлену орієнтованим графом $G=(S,V)$, вершини S якого відповідають об'єктам предметної області P , а ребра V задають відносини R між ними.

Однорідною семантичною мережею назвемо семантичну мережу, в якій всі вершини мають однаковий набір типів відносин. Неоднорідною семантичною мережею назвемо семантичну мережу, в якій вершини мають різний набір типів відношень. Іншими словами, можна сказати, що семантична мережа є однорідною, якщо всі вершини мають однакову структуру і неоднорідною - в іншому випадку.

Семантичним деревом формули назвемо зв'язну семантичну мережу без циклів $F = (S,U)$, в якій предметною областю P є множина класів математичних формул, а множиною відносин R - множина зв'язків між формулами.

Наприклад, для формули:

									Арк.
									25
Змн.	Арк.	№ докум.	Підпис	Дата	ДР.КСМ.110850/16.00.00.000 ПЗ				

$$\int_0^1 f(x)dx,$$

семантичне дерево буде мати вигляд проілюстрований на рисунку 2.2.

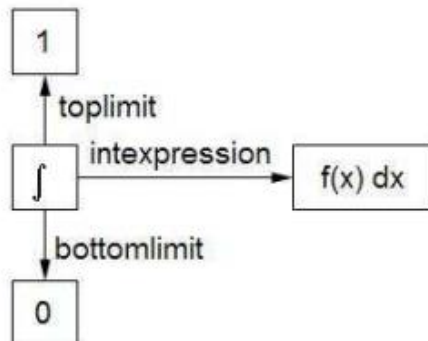


Рисунок 2.2 – Семантичне дерево для формули інтегралу

Як видно з рисунка, формула інтеграла містить вихідні ребра, що описують відношення трьох типів: нижня межа (bottomlimit), верхня межа (toplimit) і підінтегральний вираз (intexpression).

2.2 Алгоритм аналізу математичних формул

Відносини математичних операторів визначаються в залежності від позиції та відносного розміру символу в виразі. Для визначення відносин використовуються просторові області «зверху зліва», «зверху», «верхній індекс», «нижній індекс», «знизу», «знизу зліва» і «підвираз». Наприклад, очікується, що операнди (чисельник і знаменник) горизонтальної лінії (оператора дроби) будуть лежати в областях «зверху» і «знизу» відносно горизонтальної лінії.

Виходячи з простих атрибутів (верхня межа, права межа тощо), можна визначити поріг верхнього індексу, поріг нижнього індексу та інші атрибути. Ці чисельні атрибути, які використовуються для відділення областей навколо

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

символу. Також виходячи з простих атрибутів можна визначити центральну точку.

Центральна точка - це точка, яка визначає розташування символу в будь-якої з областей. Для визначення атрибутів символу, символ класифікується як з наголосами елементом, з підрядковим елементом або центральний. Кожен клас символів має по-різному визначені атрибути. Це необхідно для того, щоб уникнути неоднозначностей при структурному аналізі.

Базова лінія - це список, який представляє горизонтальне розташування символів в формулі. Кожен символ має посилання на інші базові лінії, які задовольняють різним просторовим відносинам (рисунок 2.3).

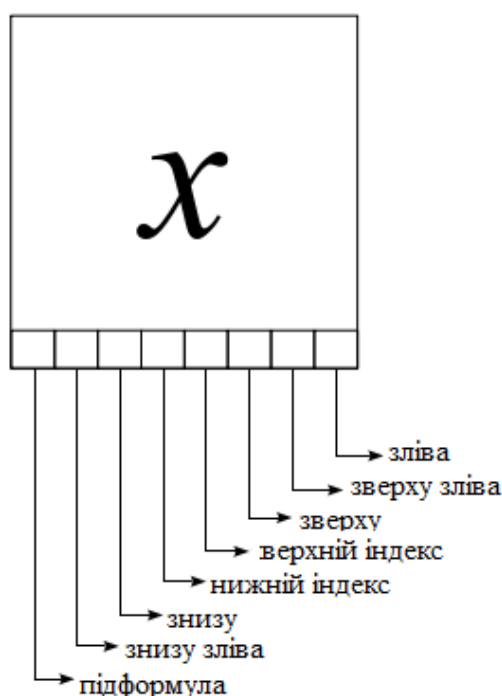


Рисунок 2.3 – Розбиття відображення символу відносно базової лінії

Домінування визначається наступним чином: символ s домінує над символом a , якщо a лежить в області s , а s не лежить в області a . Однак, обидва символи можуть лежати в областях один одного. Наприклад, символ дробу буде домінувати над символом цифри.

Розпізнавання арабських текстів відбувається статичним методом. Для цього будується матриця зображення. В отриманій матриці за допомогою побудови проєкцій відбувається пошук окремих рядків рукописного тексту.

Зазвичай будується горизонтальна проекція. Після цього виконується скелетизація зображення, тобто товщина кожної кривої стає не більше одного пікселя (одного елемента в матриці зображення). Далі відбувається отримання структури окремих компонент зв'язності в матриці. Ця структура містить інформацію про критичні точки, пронумеровані по особливому правилу; про криві, що з'єднують критичні точки; про напрямки кривих. Після цього відбувається порівняння з шаблонами. Шаблон, який набрав найбільшу кількість збігів, береться в якості вірно розпізнаного елемента тексту.

На першому етапі розпізнаються всі частини растрового зображення з першого масиву та замінюються на відповідні фрагменти математичного тексту.

На наступному етапі залишається замінити всі посилання з першого масиву на відповідні тексти другого масиву. Після цього отриманий масив перетворюється в текстовий документ, в якому повністю зберігається вихідна послідовність.

Розпізнавання виконується в кілька етапів:

1) Отримання матриці зображення. На даному етапі будується матриця, що дорівнює за розміром з вихідним зображенням. Кожен елемент матриці відповідає пікселю вихідного зображення, тобто якщо піксель зафарбований, то відповідний елемент в матриці приймає значення 1, інакше 0. Якщо зображення містить більше 2 кольорів, то обчислюється сума квітів по схемі RGB. Якщо ця сума менше фіксованого цілого числа, то елемент матриці приймає значення 1, інакше 0.

2) Побудова проекцій. Даний метод використовується для виділення окремих рядків в рукописному тексті. Для виділення рядка спочатку будується горизонтальна проекція в матриці зображення. Проекція являє собою одномірний цілочисельний масив. Кількість елементів масиву дорівнює кількості рядків в матриці. Значення елементів даного масиву виходить як сума елементів відповідного рядка матриці.

3) Стоншення зображення знаходження скелета кожного символу, тобто товщина кожної кривої в зображенні повинна залишитися рівній одному

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

пікселю або одному елементу в матриці зображення. Для отримання скелета зображення кілька разів розглядаються околиці всіх елементів. В матриці елементи, значення яких дорівнює 1. Під околом мається на увазі вісім сусідніх елементів. Якщо окіл належить множині шаблонних околів, то центральному елементу присвоюється значення 0, тобто відбувається видалення пікселя в зображенні. Стоншення відбувається в кілька етапів, щоб елементи віддалялися рівномірно з усіх боків намальованої області та залишилися тільки центральні пікселі.

4) Пошук критичних точок знайденого символу. Критичними точками є точки перетину (є більше двох сусідніх ненульових елементів) і крайні точки (є тільки один сусідній ненульовий елемент). Крім того, виділяються і проміжні точки (є два сусідніх ненульових елемента).

5) Нумерація критичних точок проводиться за допомогою циркуляційного методу. Спочатку нумеруються всі точки перетину зліва направо, потім за годинниковою стрілкою нумеруються крайні точки, пов'язані з першою точкою перетину (початок нумерації завжди з фіксованого кута), потім нумеруються крайні точки, пов'язані з другою точкою перетину. У випадку, коли скелет символу не має точок перетину, наприклад має тільки 2 крайні точки, тоді нумерація проводиться зліва направо, якщо горизонтальне відстань між точками більше ніж вертикальне, і від низу до верху в іншому випадку. Даний метод стійкий до змін довжин і кутів нахилу кривих в розпізнається символі.

б) Обчислення напрямків. Відповідно до знайдених критичних точках скелет розділяється на окремі криві, для яких обчислюються напрямки. Для цього кожна крива ділиться на 8 рівних (за кількістю елементів) частин і для кожної частини обчислюється кут нахилу, тобто кут нахилу прямої, що проходить через початок і кінець поточної частині кривої (початок і кінець частини визначаються згідно з проведеною нумерації критичних точок). Відзначимо, що криву не завжди можна розділити на 8 рівних частин, в цьому випадку довжини можуть відрізнятися на одиницю. Якщо вся крива містить

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

менше 8 елементів, тоді відбувається копіювання елементів (з одного елемента виходять 2 або більше рівні частини).

7) Порівняння з шаблоном. Порівняння із шаблоном відбувається тільки в тому випадку, якщо кількість точок перетину та кількість крайніх точок в шаблоні і поточному символі рівні і є однакові схеми їх з'єднання (згідно з нумерацією). Порівняння засноване на порівнянні напрямків. Проводиться обчислення кількості збігів. Якщо напрямки дуже близькі за значенням (наприклад відрізняються не більше ніж на 4), то вважається, що є один збіг. Якщо відмінність більше (наприклад більше ніж 4) але менше ніж), тоді вважається, що є половина збігу. Шаблон, який набрав найбільшу кількість збігів, береться в якості вірно розпізнаного символу.

8) Побудова дерева рядки. Будується дерево з індексом розгалуження 3. Є 3 гілки: bottom (поточний символ нижче попереднього), top (поточний символ вище попереднього), next (поточний символ на одному рівні з попереднім). Для побудови дерева використовуються дані, отримані на попередніх етапах. Наприклад, під час поділу на окремі символи також виробляється і обчислення розміру самого символу та його координати в оригінальному документі. Це і є підставою припущення, що символ знаходиться вище або нижче попереднього. Також використовується горизонтальна проекція, з якої вибирається так звана основна лінія рядка (елемент з максимальним значенням). Відзначимо, що в математичній формулі не завжди вдається виділити основну лінію в зв'язку зі складною структурою, наприклад якщо довжина формули невелика і вона містить кілька складних виразів на різних рівнях.

9) Обробка дерева. На даному етапі відбувається обробка окремих гілок дерева. Проводиться з'єднання декількох символів в один (наприклад "=" замінюються на знак рівності), редагування дужок, щоб вкладені дужки мали різні розміри в візуальному представленні, пошук математичних функцій тощо.

10) Побудова виразів. Для кожного типу формул будується відповідна структура. Наприклад, для інтеграла будується структура, яка містить поля верхнього і нижнього меж, подинтегрального вираження і диференціала.

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

Наприклад, для $\int_0^1 f(x)dx$ вийде наступна структура: `int.toplimit = 1`, `int.bottomlimit = 0`, `int.intexpression = f(x)`, `int.dierencial = x`. Відзначимо, що поля `toplimit`, `bottomlimit`, `intexpression` і `intexpression` так само є структурами, але вже з іншими полями, наприклад якби на місці верхньої межі інтегрування стояв x^2 , тоді `int.toplimit = toplim`, де `toplim.index = 2`, `toplim.expression = x`. Даний етап проводиться для спрощення виведення отриманих результатів.

На основі даного алгоритму розроблено спрощений алгоритм розпізнавання математичних виразів та побудови графіків даних функцій. Запропонований алгоритм складається з наступних кроків:

- отримання математичного виразу у стрічковому форматі;
- розбиття вхідної стрічки на підстрічки що містять структурні елементи обмежені спеціальними символами: дужками, дробовими ристками, знаком кореня;
- розбиття отриманих підстрічок на окремі елементи – символи;
- класифікація окремих символів на приналежність до основної лінії, перевірка чи символ не належить до верхнього чи нижнього індексу;
- виконання математичних обчислень в кожній підстрічці;
- підстановка та обчислення кінцевого значення математичного виразу шляхом послідовного аналізу підстрічок та виконання необхідних математичних операцій згідно загальнопринятих пріоритетів;
- перенесення отриманого значення на координатну сітку та промальовка частини графіка.

Даний алгоритм має ряд переваг, а саме, простоту реалізації, висока швидкість обробки даних, можливість аналізу математичних виразів різної складності.

Серед недоліків можна відзначити, що запропонований алгоритм не може опрацьовувати рукописні математичні вирази.

Проте, не зважаючи на наведені недоліки, запропонований алгоритм вирішує завдання розпізнавання математичного виразу та візуалізації графіків.

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

2.3 Моделювання програмного модуля візуалізації графіків

Розробники сучасних програмних систем постійно стикаються зі зростанням їх складності. Це пов'язано з підвищенням числа вимог до систем, застосуванням більш складних архітектурних рішень (наприклад, з переходом від простих клієнт-серверних систем до розподілених багатоланкових) і як наслідок зі збільшенням обсягів програмного коду. У цих умовах розробка системи шляхом прямого переходу від вимог до програмування призводить до величезної кількості помилок, в результаті чого такі проекти зазвичай залишаються незавершеними. Модель програмної системи створюється для того, щоб краще розуміти систему, що розробляється.

Проектування програмної системи починається з побудови моделі, яка описує систему з різних точок зору. Це означає, що повна модель складається з окремих проєкцій, що відображає різні аспекти системи. Вибір проєкцій моделі залежить від підходу до проблеми та прийнятих рішень. Ключем до побудови проєкцій є абстрагування, коли зосереджуються на найбільш істотних деталях, ігноруючи при цьому менш важливі. Тобто, кожна проєкція – це модель. Наприклад, функціональна модель показує, що робить система у відповідь на запити користувача (але не як вона це робить), логічна модель описує основні сутності і відносини між ними.

Моделі дозволяють звести високу складність програмної системи до рівня, що розуміється людиною. Досягається це за рахунок ієрархічного принципу їх побудови та застосування наочної графічної нотації. Ієрархія рівнів опису системи дає можливість різко скоротити кількість елементів, які повинен аналізувати розробник. Кожна модель може бути виражена з різними рівнями деталізації. При цьому на верхніх рівнях ієрархії опускаються деталі реалізації, які проявляються на більш низьких рівнях.

Головна перевага моделей - подання інформації в наочній графічній формі. Для однакового розуміння графічного представлення різними людьми графічна нотація повинна бути уніфікована, тобто повинна використовуватися

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

мова, що визначає таку нотацію і її семантику. Саме такою і є Unified Modeling Language (UML).

Для отримання цілісного уявлення про роботу окремих елементів системи було розроблено діаграму прецедентів програмного додатку малювання графіків довільної складності. Приклад діаграми прецедентів наведено на рисунку 2.4.

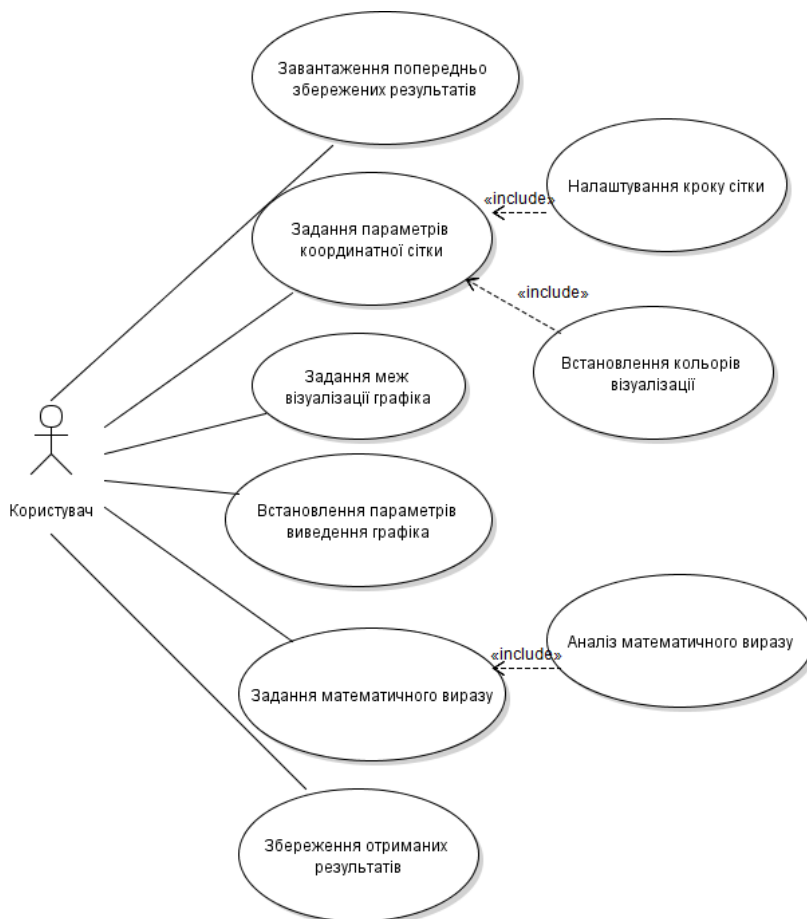


Рисунок 2.4 – Діаграма прецедентів програмного додатку візуалізації математичних виразів

При аналізі технічного завдання було виділено одну групу користувачів, що будуть мати доступ до всіх функціональних можливостей програмного модуля. Серед задач, які можуть вирішуватись даною групою є:

- створення/завантаження/збереження зображень графіків згідно заданого математичного виразу;
- встановлення параметрів роботи програмного модуля;
- встановлення параметрів координатної сітки;

- можливості масштабування отриманих графіків;
- встановлення підписів під осями графіків;
- друк отриманих результатів роботи програми.

Даний набір функцій в повній мірі дозволяє виконати усі поставлені завдання та отримати візуальне представлення математичного виразу у графічному вигляді на заданому діапазоні значень.

Для перевірки коректної взаємодії між користувачем та окремими програмними модулями додатково було проведено моделювання за допомогою діаграми послідовностей. Результати моделювання наведено на рисунку 2.5.

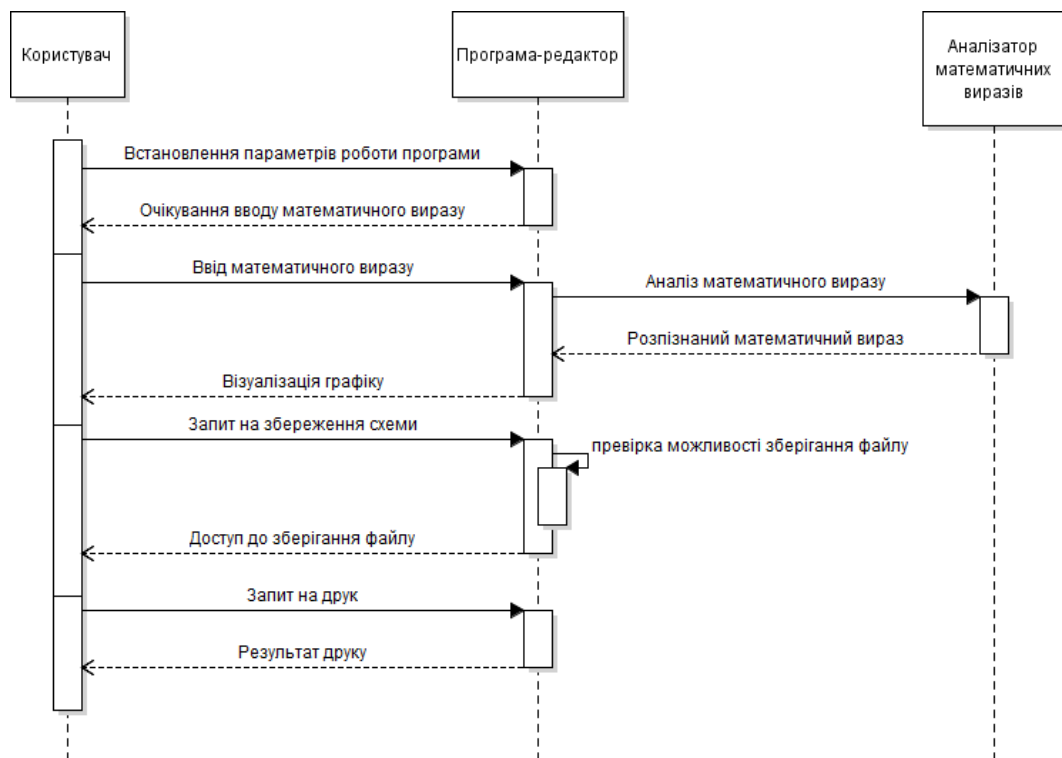


Рисунок 2.5 – Діаграма послідовностей програмного додатку візуалізації математичних виразів

При алалізі можливиз запитів між користувачем та окремими складовими системи помилок не виявлено, що доводить правильність обраної моделю при проектування програмного модуля.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОГРАМНОГО МОДУЛЯ ВІЗУАЛІЗАЦІЇ ГРАФІКІВ МАТЕМАТИЧНИХ ВИРАЗІВ

3.1 Опис структури програмного додатку

Для написання програмного модуля візуалізації математичних виразів було обрано програмне середовище Delphi версії 7 фірми Enterprise (Borland), оскільки воно надає найбільш широкі можливості для програмування додатків для ОС Windows. Delphi - це продукт Borland International для швидкого створення додатків. RAD-середовища ще називають візуальними середовищами розробки при використанні яких розробники бачать робочі та діалогові вікна програми при проектуванні, такими вони й будуть, коли програма запрацює.

Високопродуктивний інструмент візуального побудови додатків включає в себе справжній компілятор коду та надає засоби візуального програмування, подібні на Microsoft Visual Basic або в інші інструменти візуального проектування. В основі Delphi лежить мова Object Pascal, який є розширенням об'єктно-орієнтованої мови Pascal. У Delphi також входять локальний SQL-сервер, генератори звітів, бібліотеки візуальних компонентів, і інше, необхідне для того, щоб відчувати себе абсолютно впевненим при професійній розробці інформаційних систем або просто програм для Windows-середовища.

Перш за все Delphi призначений для професійних розробників, бажаючих дуже швидко розробляти програми в архітектурі клієнт-сервер. Delphi виробляє невеликі за розмірами високоефективні виконавчі модулі (.exe і .dll), тому в Delphi повинні бути, перш за все, зацікавлені ті, хто розробляє продукти на продаж. З іншого боку невеликі за розмірами і швидко виконувані модулі означають, що вимоги до клієнтських робочих місць істотно знижуються - це має важливе значення і для кінцевих користувачів.

Переваги Delphi в порівнянні з аналогічними програмними продуктами:

- швидкість розробки програми (RAD);
- висока продуктивність розробленого додатка;
- низькі вимоги розробленого додатка до ресурсів комп'ютера;

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

- масштабованість за рахунок вбудовування нових компонент і інструментів в середу Delphi;
- можливість розробки нових компонентів і інструментів власними засобами Delphi;
- наявність програмних ресурсів для обробки графічних елементів.

Система програмування Delphi розрахована на програмування різних додатків і надає велику кількість компонентів для цього. Можливості Delphi повністю відповідають цим вимогам і підходять для створення систем будь-якої складності.

На етапі проектування була запропонована наступна структура програмного модуля (рисунок 3.1):

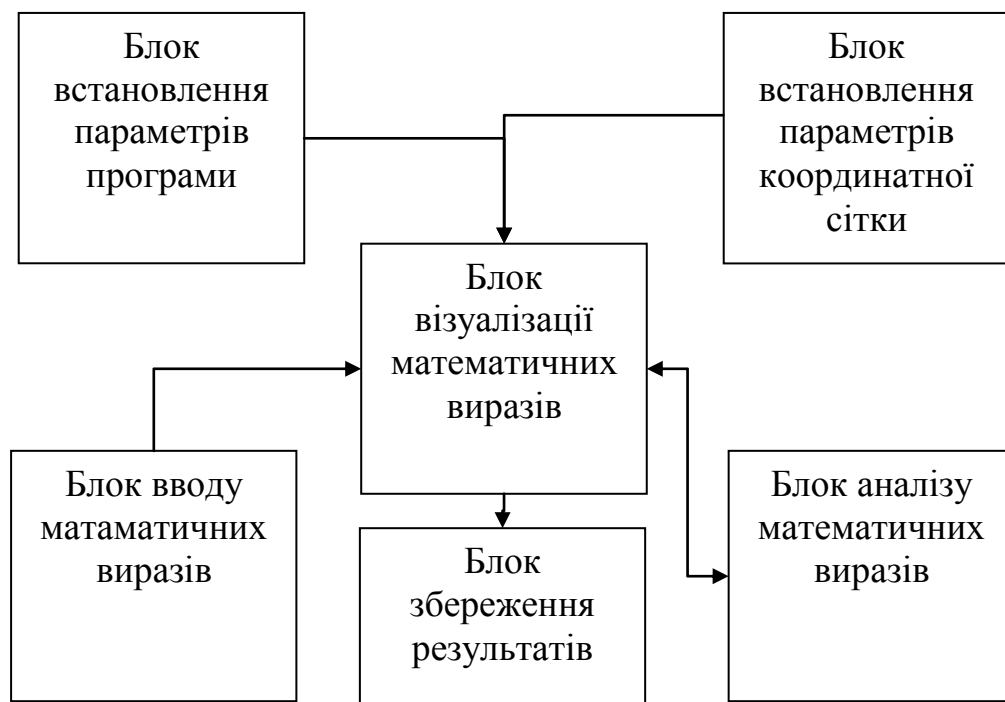


Рисунок 3.1 – Структура програмного модуля візуалізації складних математичних графіків

Розроблена структура дозволяє вносити зміни в кінцевий програмний модуль без особливих втручань в роботоздатність цілого модуля. Розглянемо функціональні можливості кожного структурного елемента програмного модуля.

“Блок встановлення параметрів програми”- містить функції, призначені для встановлення початкових параметрів роботи програми, відповідає з перевірку цілісності програмного модуля та можливість подальшого коректного функціонування.

“Блок встановлення параметрів координатної сітки” – включає функції корекції параметрів відображення графіків, координатної сітки в залежності від поставлених завдань. Серед основних параметрів є: підписи під осями абсцис та ординат, задана кроку маштабуючої лінійки, встановлення кольору графіка при візуалізації, встановлення меж графіку тощо. Даний блок містить функції для підвищення естетичного сприйняття отриманих графіків.

“Блок візуалізації математичних виразів” – об’єднує програмні функції для відображення масиву точок, які були отримані під час обчислення математичних виразів у зручному для аналізу форматі. Результуючий графік відображається на головному вікні програмного модуля та може бути збереженим у зручному для користувача форматі.

“Блок вводу математичних виразів” – відповідає за введення/завантаження математичних виразів для подальшого опрацювання. Під час задання математичних виразів слід використовувати ряд спеціалізованих скорочень та символів. Для запису математичних функцій:

- Abs. Функція модуля. Наприклад: $Abs(x^2-x)$, результат: $|x^{2-x}|$;
- Sqr. Зведення в квадрат. Наприклад: $Sqr(x)$, аналог $x*x$ або x^2 ;
- Sqrt. Витяг квадратного кореня. Наприклад: $Sqrt(4)$, результат дорівнює 2;
- Exp. Функція експоненти;
- Ln. Функція натурального логарифма;
- Lg. Функція десяткового логарифма;
- Log. Функція логарифма з довільною основою.

Диференціали та інтеграли:

- Dx. Похідна функції по аргументу x. Наприклад: $Dx(x^3+y)$, результат: $3x^2$;
- Dy. Похідна функції по аргументу y;

									Арк.
									37
Змн.	Арк.	№ докум.	Підпис	Дата	ДР.КСМ.110850/16.00.00.000 ПЗ				

- Integral. Інтегрування функції методом Сімпсона;
- Integral3D. Інтегрування функції багатьох змінних методом Монте-Карло.

Тригонометричні функції:

- Sin, Cos, Tg, Ctg, Sec, CoSec. Тригонометричні функції синуса, косинуса, тангенса, котангенса, секанса і косеканс.

Зворотні тригонометричні функції:

- ArcSin, ArcCos, ArcTg, ArcCtg, ArcSec, ArcCoSec. Зворотні тригонометричні функції синуса, косинуса, тангенса, котангенса, секанса і косеканс.

Гіперболічні функції:

- Sh, Ch, Th, Cth, Sch, CSch. Гіперболічні функції синуса, косинуса, тангенса, котангенса, секанса і косеканс.

Зворотні гіперболічні функції:

- ArSh, ArCh, ArTh, ArCth, ArSch, ArCSch. Зворотні гіперболічні функції синуса, косинуса, тангенса, котангенса, секанса і косеканс.

Результатом роботи даного модуля є стрічкова змінна, що містить деякий математичний вираз який необхідно графічно відобразити на заданому проміжку.

“Блок аналізу математичних виразів” – основний структурний елемент програмного модуля в якому відбувається розпізнавання та подальше обчислення математичного виразу з метою отримання множини точок. Набір точок є базою для побудови графіку функції.

“Блок збереження результатів” – допоміжний елемент системи, що дозволяє зберігати на зовнішніх носіях даних результатів роботи реалізованого програмного модуля.

Запропонована структура розбиває поставлену задачу виводу графіків функцій на окремі взаємо не пов’язні підзадачі. Даний підхід дозволив значно спростити процес програмування й тестування окремих функцій та модулів. В результаті реалізації розробленої структури було отримано програмний модуль візуалізації складних математичних графіків.

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

Для простоти експлуатації та швидкого процесу навчання було спроектовано та реалізовано графічний інтерфейс користувача, який наведено на рисунку 3.2.

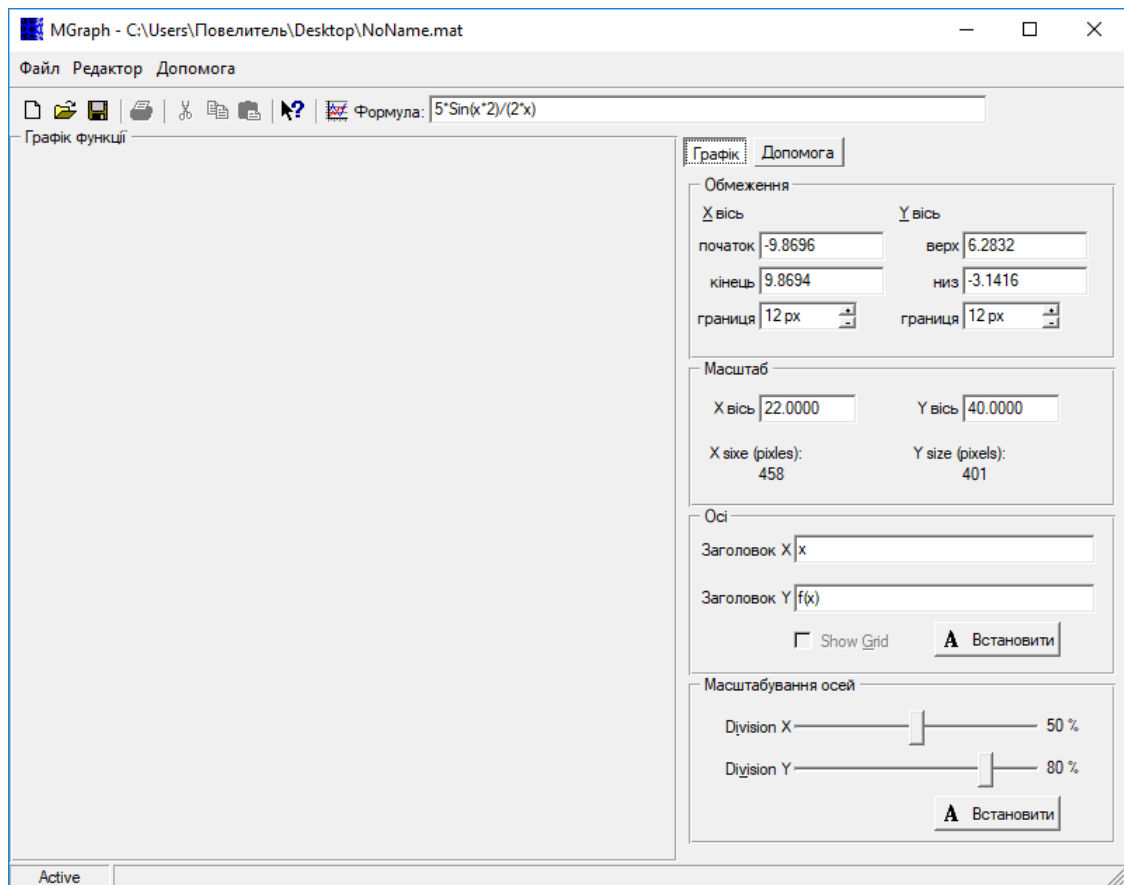


Рисунок 3.2 – Головне вікно програмного модуля

На головному вікні розміщені функціональні перемикачі та активатори за допомогою яких можна отримати доступ до всіх можливих ресурсів системи. Для зручності перегляду результатів роботи програми користувачеві надається графічне вікно в якому буде відображатись графік функції, що була передана як вхідна стрічка. Окрім того, всі керуючі елементи для встановлення параметрів роботи програми додатково було винесено на головне вікно програмного додатку. Дане дизайнерське рішення було прийнято для зменшення часу налаштування програми для роботи користувача з відповідним математичним виразом. Окрім того, в дизайні графічного інтерфейсу користувача було використано візуальну подібність до прикладних програм, які розробляються для операційної системи Windows, що значно спрощує опанування програмою.

						Арк.
					ДР.КСМ.110850/16.00.00.000 ПЗ	39
Змн.	Арк.	№ докум.	Підпис	Дата		

3.2 Реалізація методів розпізнавання та візуалізації математичних виразів

Основною задачею при роботі даного програмного модуля є процес розпізнавання вхідного математичного виразу, оскільки саме від цього етапу в основному залежить остаточний результат роботи програми. Тому розглянемо реалізацію даного етапу більш детально.

На вхід даного структурного блоку передається стрічкова інформація, яка містить в собі послідовність спеціалізованих символів для задання математичного виразу, що описуватиме графік. Якщо є упорядкований список символів L , то можна визначити крайній лівий домінуючий символ в списку шляхом виконання наступної рекурсивної процедури:

`getDominantSymbol(L):`

Нехай $n = \text{length}(L)$ - кількість символів в списку.

Якщо $n == 1$, повернути $s(n)$

Якщо $s(n)$ домінує над $s(n-1)$, видалити $s(n-1)$ з L , інакше видалити $s(n)$

Повернути `getDominantSymbol(L)`.

Домінуюча базова лінія формули - це базова лінія, на яку не посилається ніякий символ. Під час розпізнавання математичного виразу зазвичай спочатку проводиться пошук крайнього лівого домінуючого символу, потім наступного за ним домінуючого символу, і так далі, поки не будуть знайдені всі символи базової лінії.

Структура даних, яка представляє вираз, називається деревом базових ліній. Наприклад, для формули:

$$x_{ij} \cdot y + \frac{a+b}{c},$$

дерево базових ліній цього виразу матиме такий вигляд (рисунок 3.3):

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

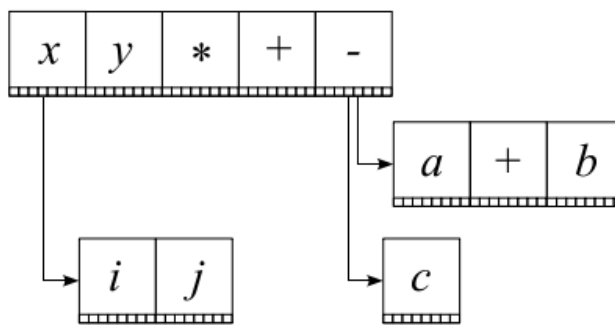


Рисунок 3.3 – Дерево базових ліній

Домінуючу базову лінію Db конструємо за допомогою наступної функції $\text{constructDominantBaseline}(Db, L)$:

Якщо Db порожньо, то $Db = \text{addSymbol}(Db, \text{getDominantSymbol}(L))$

$s = \text{getLastSymbol}(Db)$

Створити список $Hs = \text{getRightNeighbors}(s, L)$ символів з L , які є правими горизонтальними сусідами символу s

Якщо Hs порожній, вийти

Знайти домінуючий символ серед символів з Hs , $sd = \text{getDominantSymbol}(Hs)$

$Db = \text{addSymbol}(Db, sd)$

Рекурсія: $\text{constructDominantBaseline}(Db, L)$

Дерево базових ліній створюється рекурсивним знаходженням домінуючих базових ліній в формулі, описаному впорядкованим списком символів L . $\text{ConstructBaselineTree}(L)$:

якщо L порожній, вихід;

ініціалізація Db ;

$\text{constructDominantBaseline}(Db, L)$;

оновити Db , згрупувавши символи, що визначають імена функцій і операторів

Для кожного символу s з Db створити нові списки символів з нащадками, в залежності від того, яким просторовим відносинам вони задовольняють, і створити посилання на ці списки;

$L = Db$;

Для кожного символу використовувати рекурсію, застосовуючи процедуру constructBaselineTree до кожного зі списків-нащадків.

Приклад реалізації функції відображення графіків на головному вікні програми наведено нище:

```
Self.imgGraphic.Canvas.Pen.Color := clNavy; flgPaint:=false;
if MathematicsFunc.Parse_Line(Func, workfunc) then
begin for i:=Self.FGraphicParam.Xborder to Self.FGraphicParam.Xborder +
Self.FGraphicParam.PaintX
do begin
j := Self.FGraphicParam.Xmin+(Self.FGraphicParam.Xmax-Self.FGraphicParam.
Xmin) *((i-Self.FGraphicParam.Xborder) /Self.FGraphicParam. PaintX) ;
if not MathematicsFunc.Count_Func(true, workfunc, j, 0, funcresult) then begin
Self.imgGraphic.Canvas.MoveTo(i, Xaxis); flgPaint:=false; Continue; end;
funcresult := -funcresult*Self.FGraphicParam.DivY+ Self.FGraphicParam. Ymax
Self.FGraphicParam.DivY+Self.FGraphicParam.Yborder;
```

За допомогою даного програмного коду проводиться розмітка майдутнього графіка та перевід маркера виведення точок графіка на відповідні місця. Для відображення використовується стандартне полотно виводу графічної інформації (canvas), що дозволить в подальшому використати всі моливості середовища Delphi для роботи з графічними даними. Після виокнання даної частини коду отримується підготовлене полотно для вивдення графіка функції.

```
if i=Self.FGraphicParam.Xborder
then Self.imgGraphic.Canvas.MoveTo(i, Round(funcresult))
else if (Round(funcresult)<Self.FGraphicParam.Yborder) then
Self.imgGraphic.Canvas.MoveTo(i, Self.FGraphicParam.Yborder)
```

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

```

else if (Round(funcresult) > Self.FGraphicParam.Yborder+
Self.FGraphicParam.PaintY) then Self.imgGraphic.Canvas.MoveTo (i,
Self.FGraphicParam.Yborder +Self.FGraphicParam.PaintY)
else if flgPaint then Self.imgGraphic.Canvas.LineTo(i, Round(funcresult)) else
Self.imgGraphic.Canvas.MoveTo(i, Round(funcresult));
flgPaint:=true;end;MathematicsFunc.Delete_Tree(workfunc); end;

```

Дана частина коду відповідає за саму візуалізацію графічного відображення математичної функції на заданому проміжку, та з заданими параметрами. Оскільки зображення на екрані складається з пікселів, а результати обчислення математичної функції не завжди є цілим числом, то проводиться процедура округлення до найблищого цілого. Дане округлення вносить деяке спотворення в загальний вигляд графіків, проте при використанні великих значень масштабів похибка візуально мінімізується.

3.3 Тестування програмного модуля візуалізації складних графіків

Для проведення тестування розробленого програмного модуля було вибрано ПК з наступними технічними характеристиками:

- процесор - Intel Core i5;
- модель - 7400 (Intel® Virtualization Technology);
- частота 3.0 ГГц;
- оперативна пам'ять 8 Гб;
- тип встановленої пам'яті - DDR4 (DIMM);
- частота встановленої пам'яті - 2400 мгц (pc4-19200);
- HDD 1000 гб;
- блок живлення 400 Вт.

Характеристики монітору:

- діагональ 23";

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

- тип матриці IPS;
- максимальна роздільна здатність 1920x1080 (Full HD);
- час відгуку матриці 5 мс (сірий до сірого);
- яскравість 250 кд / м²;
- контрастність статична 1000: 1;
- інтерфейси підключення 1xVGA(D-Sub), 1xDVI-D, 1xHDMI.

При проведенні тестування було проведено ряд тестів з формулами різної структурної складності, наприклад для формули:

$$f(x) = \frac{5 * \sin x^2}{2 * x}$$

Графік функції буде мати наступний вигляд (рисунок 3.4):

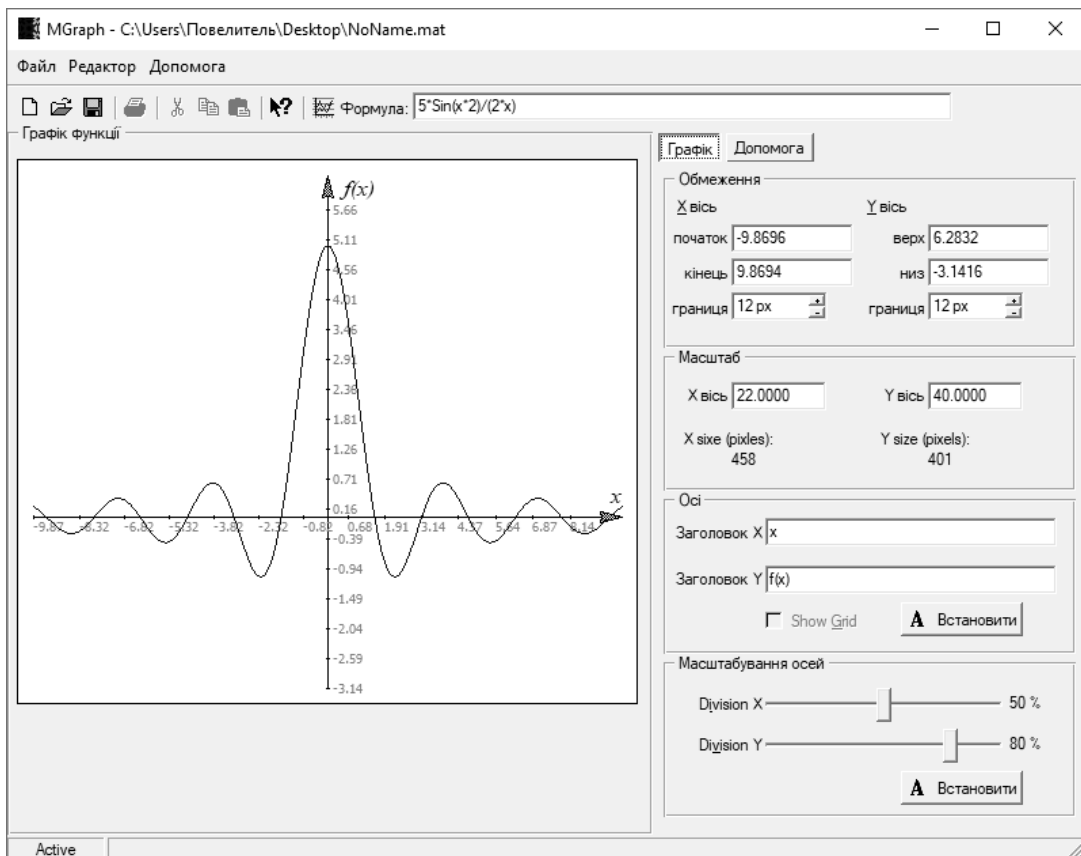
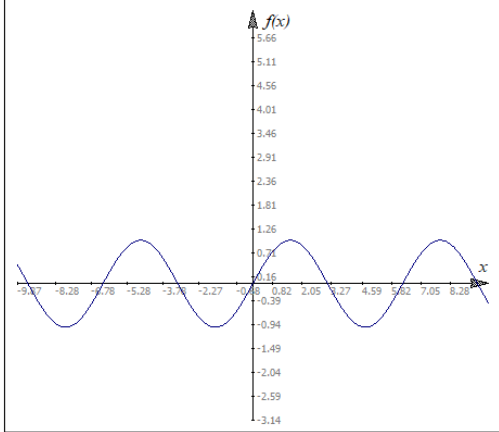
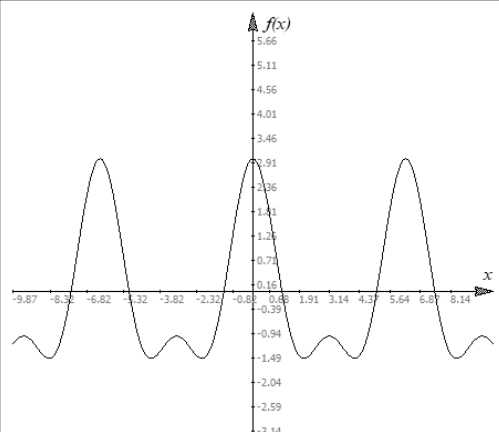
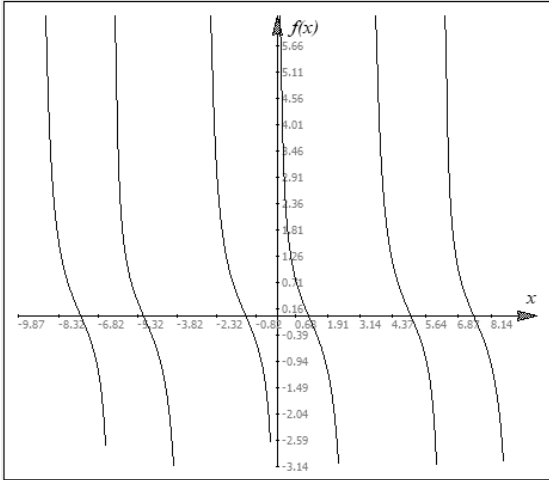


Рисунок 3.4 – Приклад роботи розробленого рограмного модуля

Результати тестування графіків, що задаються іншими типами математичних формул наведено в таблиці 3.1.

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

Таблиця 3.1 – Приклади візуалізації графіків різної складності

Функція	Графік
$f(x) = \sin(x)$	
$f(x) = 2 * \cos(x) + \cos(2x)$	
$f(x) = \frac{2 * \cos(x) + \cos(2x)}{2 * \sin(x) + \sin(2x)}$	

За результатами проведених тестувань можна зробити наступні висновки:

- розроблений програмний модуль вірно розпізнає формули довільної складності;
- аналіз математичних виразів відбувається без помітних затримок;
- графіки функцій відображаються у зручному та наочному вигляді.

Змн.	Арк.	№ докум.	Підпис	Дата

4 ТЕХНІКО-ЕКОНОМІЧНИЙ РОЗДІЛ

Метою техніко – економічного розділу дипломної роботи є здійснення економічних розрахунків, спрямованих на визначення економічної ефективності програмного модуля розпізнавання та візуалізації математичних виразів довільної складності та прийняття рішення про його подальший розвиток і впровадження або ж недоцільність проведення відповідної розробки. Для проведення даного дослідження необхідно провести ряд розрахунків.

4.1 Розрахунок витрат на розробку програмного додатку

Витрати на розробку і впровадження програмного модуля візуалізації графіків (К) включають:

$$K = K_1 + K_2,$$

де K_1 - витрати на розробку апаратного та програмного забезпечення грн.;

K_2 - витрати на відлагодження і дослідну експлуатацію програми рішення задачі на комп'ютері, грн.

Витрати на розробку апаратних та програмних засобів включають:

- витрати на оплату праці розробників (V_{OP});
- витрати на відрахування у спеціальні державні фонди (V_{ϕ});
- витрати на матеріали та комплектуючі (Π_b);
- накладні витрати (Н);
- інші витрати (I_b)
- витрати на використання комп'ютерної техніки (V_{KT}) .

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

Розрахунок витрат на оплату праці.

Витрати на оплату праці включають заробітну плату (ЗП) всіх категорій працівників, безпосередньо зайнятих на всіх етапах проектування. Розмір ЗП обчислюється на основі трудоемності відповідних робіт у людино-днях та середньої ЗП відповідних категорій працівників.

У розробці проектного рішення задіяні наступні спеціалісти - розробники, а саме: керівник проекту; студент-дипломант; консультант техніко-економічного розділу (Таблиця 4.1).

Таблиця 4.1 - Вихідні дані для розрахунку витрат на оплату праці

№п/п	Посада виконавців	Місячний оклад, грн.
1	Керівник ДП, викладач	5286
2	Консультант техніко-економічного розділу, доцент	5656
3	Студент	1200

Витрати на оплату праці розробників проекту визначаються за формулою (4.1):

$$B_{ОП} = \sum_{i=1}^N \sum_{j=1}^M n_{ij} \cdot t_{ij} \cdot C_{ij} , \quad (4.1)$$

де n_{ij} – чисельність розробників i -ої спеціальності j -го тарифного розряду, осіб;

t_{ij} – затрачений час на розробку проекту співробітником i -ої спеціальності j -го тарифного розряду, год;

C_{ij} – годинна ставка працівника i -ої спеціальності j -го тарифного розряду, грн.,

Середньо годинна ставка працівника може бути розрахована за формулою (4.2):

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

$$C_{ij} = \frac{C_{ij}^0(1+h)}{PЧ_i}, \quad (4.2)$$

де C_{ij} – основна місячна заробітна плата розробника і-ої спеціальності j-го тарифного розряду, грн.;

h – коефіцієнт, що визначає розмір додаткової заробітної плати (при умові наявності доплат);

$PЧ_i$ - місячний фонд робочого часу працівника і-ої спеціальності j-го тарифного розряду, год. (приймаємо 168 год.).

Результати розрахунку записують до таблиці 4.2.

Таблиця 4.2 - Розрахунок витрат на оплату праці

№ п/п	Посада виконавців	Час розробки, год	Погодинна заробітна плата, грн/год.	Витрати на розробку, грн
1	Керівник ДП, старший викладач	20,5	36,03	738,62
2	Консультант техніко-економічного розділу, доцент	2	38,42	76,84
3	Студент	144	6,55	943,2
Разом				1758,66

Відрахування на соціальні заходи.

Величину відрахувань у спеціальні державні фонди визначають у відсотковому співвідношенні від суми основної та додаткової заробітних плат. Згідно діючого нормативного законодавства сума відрахувань у спеціальні державні фонди складає 16,4% від суми заробітної плати:

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

$$B_{\Phi} = \frac{16,4}{100} \cdot 1758,66 = 288,42 \text{ грн.}$$

Розрахунок витрат на матеріали та комплектуючі.

Загальна сума витрат на матеріальні ресурси (B_M) визначається за формулою (4.3):

$$B_M = \sum_{i=1}^n K_i \cdot C_i, \quad (4.3)$$

де K_i - витрата i -го типу матеріалу, натуральні одиниці вимірювання;

C_i - ціна за одиницю i -го типу матеріалу, грн.;

i - тип матеріального ресурсу;

n - кількість типів матеріальних ресурсів.

Таблиця 4.3 - Зведені розрахунки матеріальних витрат

№ п/п	Найменування матеріальних ресурсів	Од. виміру	Факт. витрачено матеріалів	Ціна за одиницю, грн.	Сума, грн	Транспортні витрати (10% від суми)	Загальна сума, грн	
	Допоміжна література	шт	1	600	600	60	660	
	Папір (формат А4)	уп	2	70	140	14	154	
	Ручка кулькова	шт	2	10	20	2	22	
	Олівець простий	шт	2	10	20	2	22	
	Диски CD-R	шт	2	10	20	2	22	
	Зошит, 96 арк	шт	1	50	50	5	55	
	Тонер для принтера	уп	1	70	70	7	77	
Р а з о м								1012,00

Витрати на використання комп'ютерної техніки.

Витрати на використання комп'ютерної техніки (V_{KT}) включають витрати на амортизацію комп'ютерної техніки, витрати на користування програмним забезпеченням, витрати на електроенергію, що споживається комп'ютером. За даними обчислювального центру ТНЕУ для комп'ютера типу IBM PC/ATX вартість години роботи становить 6 грн. Середній щоденний час роботи на комп'ютері – 2 години. Розрахунок витрат на використання комп'ютерної техніки приведений в таблиці 4.4.

Таблиця 4.4- Розрахунок витрат на використання комп'ютерної техніки

№ п/п	Назва етапів робіт, при виконанні яких використовується комп'ютер	Час використання комп'ютера, год.	Витрати на використання комп'ютера грн.
1	Проведення досліджень та оформлення їх результатів	60	360
2	Оформлення техніко-економічного розділу	8	48
3	Оформлення ДП	12	72
Разом		80	480

Накладні витрати.

Накладні витрати проектних організацій включають три групи видатків: витрати на управління, загальногосподарські витрати, невиробничі витрати. Вони розраховуються за встановленими відсотками до витрат на оплату праці. Середньостатистичний відсоток накладних витрат приймемо 150% від заробітної плати:

$$H = 1,5 \cdot 1758,66 = 2637,99 \text{ (грн).}$$

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

Інші витрати.

Інші витрати є витратами, які не враховані в попередніх статтях. Вони становлять 10% від заробітної плати:

$$I_B = 1758,66 \cdot 0,1 = 175,87 \text{ (грн).}$$

Витрати на розробку програмного забезпечення складають:

$$K_1 = V_{OP} + V_{\Phi} + V_M + H + I_B + V_{KT},$$

$$K_1 = 1758,66 + 288,42 + 1012,00 + 2637,99 + 175,87 + 480,00 = 6352,94 \text{ (грн).}$$

Витрати на відлагодження і дослідну експлуатацію програмного продукту визначаємо за формулою (4.4):

$$K_2 = S_{m.g.} \cdot t_{eid} \quad (4.4)$$

де $S_{m.g.}$ - вартість однієї машино-години роботи ПК, грн./год;

t_{eid} - комп'ютерний час, витрачений на відлагодження і дослідну експлуатацію створеного програмного продукту, год.

Загальна кількість днів роботи на комп'ютері дорівнює 30 днів. Середній щоденний час роботи на комп'ютері – 2 години. Вартість години роботи комп'ютера дорівнює 6 грн., тому $K_2 = 6 \cdot 60 = 360$ грн.

4.2 Визначення експлуатаційних витрат

Для оцінки економічної ефективності розроблювальної системи моніторингу слід порівняти її з аналогом, тобто існуючим програмним забезпеченням ідентичного функціонального призначення.

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

Експлуатаційні одноразові витрати по програмному забезпеченню і аналогу включають вартість підготовки даних і вартість роботи комп'ютера (за час дії програми):

$$E_{\Pi} = E_{1\Pi} + E_{2\Pi},$$

де E_{Π} - одноразові експлуатаційні витрати на ПЗ (аналог), грн.;

$E_{1\Pi}$ - вартість підготовки даних для експлуатації ПЗ (аналог), грн.;

$E_{2\Pi}$ - вартість роботи комп'ютера для виконання проектного рішення (аналог), грн.

Річні експлуатаційні витрати $V_{E\Pi}$ визначаються за формулою:

$$V_{E\Pi} = E_{\Pi} * N_{\Pi},$$

де N_{Π} - періодичність експлуатації ПЗ (аналог), раз/рік.

Вартість підготовки даних для роботи на комп'ютері визначається за формулою:

$$E_{1\Pi} = \sum_{l=1}^n n_i t_i c_i,$$

де i - категорії працівників, які приймають участь у підготовці даних ($i=1,2,\dots,n$);

n_i - кількість працівників i -ої категорії, осіб.;

t_i - трудомісткість роботи співробітників i -ої категорії по підготовці даних, год.;

c_i - середнього годинна ставка працівника i -ої категорії з врахуванням додаткової заробітної плати, що знаходиться із співвідношення:

$$c_i = \frac{c_i^0 (1+b)}{m},$$

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

де c_i^0 - основна місячна заробітна плата працівника і-ої категорії, грн.;

b - коефіцієнт, який враховує додаткову заробітну плату (прийmemo 0,57);

m - кількість робочих годин у місяці, год.

Для роботи з даними як для проектного рішення так і аналогу потрібен один працівник, основна місячна заробітна плата якого складає: $c = 3200$ грн.

Тоді:

$$c_1 = \frac{3200(1 + 0,57)}{22 * 8} = 28,55 \text{ грн/год}$$

Трудомісткість підготовки даних для проектного рішення складає 1 год., для аналога 1,5 год.

Таблиця 4.5- Розрахунок витрат на підготовку даних та реалізацію проектного рішення на комп'ютері

№	Час роботи співробітників, год.	Середньогодинна заробітна плата, грн./год.	Витрати, грн.
	Проектне рішення		
1	1	28,55	28,55
	Аналог		
1	1,5	28,55	42,83

Витрати на експлуатацію комп'ютера визначається за формулою:

$$E_{2\Pi} = t * S_{MG}$$

де t - витрати машинного часу для реалізації рішення (аналогу), год.;

S_{MG} - вартість однієї години роботи комп'ютера, грн./год.

$$E_{2\Pi} = 1 * 6 = 6 \text{ грн.}; E_{2A} = 1,5 * 6 = 9 \text{ грн.}$$

$$E_{\Pi} = 28,55 + 6 = 34,55 \text{ грн.}; E_A = 42,83 + 9 = 51,83 \text{ грн.}$$

$$B_{E\Pi} = 34,55 * 252 = 8706,6 \text{ грн.}; B_{EA} = 51,83 * 252 = 13061,16 \text{ грн.}$$

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

Обчислення накладних витрат.

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства (фірми) та створення необхідних умов праці.

В залежності від організаційно-правової форми діяльності господарюючого суб'єкта, накладні витрати можуть становити 60–100 % від суми основної та додаткової заробітної плати працівників.

$$H_B = 0,7 * V_{OP}, \quad (4.7)$$

де H_B – накладні витрати.

$$H_B = 0,7 * 5024,00 = 3516,8 \text{ грн.}$$

Складання кошторису витрат та визначення собівартості.

Результати проведених розрахунків зведемо у таблицю 4.6.

Таблиця 4.6 - Кошторис витрат

№ п/п	Найменування витрат	Сума витрат, грн.
1	Витрати на оплату праці	1758,66
2	Відрахування у спеціальні державні фонди	288,42
3	Витрати на матеріали та комплектуючі	1012,00
4	Накладні витрати на розробку	2637,99
5	Інші витрати	175,87
6	Витрати на відлагодження і дослідну експлуатацію програмного продукту	360
7	Накладні витрати експлуатацію	3516,8
8	Річні експлуатаційні витрати	13061,16
Разом		22450,9

Розрахунок ціни проекту.

Договірна ціна (C_D) для проектних рішень розраховується за формулою (4.8):

$$C_D = B_{KC} \cdot \left(1 + \frac{p}{100}\right), \quad (4.8)$$

де B_{KC} – кошторисна вартість, грн.;

p - середній рівень рентабельності, % (приймаємо 30% за погодженням з керівником).

$$C_D = 22450,9 \cdot (1 + 0,3) = 29186,17 \text{ грн.}$$

4.3 Визначення економічної ефективності і терміну окупності капітальних вкладень

Економічна ефективність (E_P) полягає у відношенні результату виробництва до затрачених ресурсів:

$$E_P = \frac{\Pi}{B_{KC}}, \quad (4.9)$$

де Π – прибуток, грн.;

B_{KC} – кошторисна вартість, грн..

$$E_P = 6735,27 \text{ грн.} / 22450,9 \text{ грн.} = 0,3.$$

Поряд із економічною ефективністю розраховують термін окупності капітальних вкладень (T_p):

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

$$T_P = \frac{1}{E_P} . \quad (4.10)$$

Тобто: $T_P = 1/0,3 = 3,4$ р.

Прийнятним вважається термін окупності близький до 7 років.

Розраховані економічні показники проекту занесемо до таблиці 4.7.

Таблиця 4.7 - Економічні показники розробки

№ п/п	Показник	Значення
1.	Собівартість, грн.	22450,9
2.	Плановий прибуток, грн.	6735,27
3.	Ціна, грн.	29186,17
4.	Економічна ефективність	0,3
5.	Термін окупності, рік	3,4

Враховуючи основні економічні показники з таблиці 4.7, можна зробити висновок, що при економічній ефективності 0,3 та терміні окупності – 3,4 роки проводити роботи по впровадженню даного програмного є доцільним та економічно вигідним.

ВИСНОВКИ

На основі аналізу сучасних програмних додатків аналізу математичних виразів та візуалізації формул можна зробити наступні висновки:

1. Проведено аналіз підходів до задання математичних виразів, на основі дослідження специфікацій які описують способи запису формул, що дозволило виділити основні завдання та особливості запису математичних формул.

2. Здійснено огляд та класифікацію існуючих типів графіків, на основі сфер застосування кожного типів графіків, що дозволило довести найбільшу поширеність плоских графіків для візуалізації та табличних або формульних даних.

3. Проведено аналіз сучасних програмних додатків, які дозволяють розпізнавати математичні вирази та візуалізувати графіки, що дозволило визначити основні вимоги які ставляться перед розробниками.

4. Розроблено алгоритм аналізу математичних виразів на основі структурного підходу та системи спеціалізованих позначень, що дозволило провести розпізнавання математичних виразів довільної складності.

5. Спроектовано структуру програмного модуля візуалізації математичних формул довільної складності, на основі модульного підходу, що дозволило провести попереднє дослідження запропонованого рішення за допомогою технологій моделювання.

6. Реалізовано програмний модуль візуалізації математичних формул та проведено тестування розробленого програмного забезпечення при розпізнаванні формул різної складності.

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Баженова И.Ю. Язык программирования Delphi// И.Ю. Баженова/- АО "Диалог-МИФИ", 1997. – 366с.
2. Бартлетт Н. Программирование на Delphi Путеводитель // Н. Бартлетт А. Лесли, С. Симкин /.- The Cogniolis Group,Inc.,1996, Издательство НИПФ "ДиаСофт Лтд.",1996. – 116с.
3. Вебер Дж. Технология С++в подлиннике // Дж. Вебер /- QUE Corporation, 1996, "ВНУ-Санкт-Петербург",1997. – 256с.
4. Волш А. И. Основы программирования на С++для World Wide Web // А. И. Волш /- IDG Books Worldwide,Inc.,1996, - Издательство "Диалектика",1996. – 458с.
5. Марков А. С. «Базы данных. Введение в теорию и методологию // А. С. Марков, К.Ю. Лисовский / - Финансы и статистика»-2006,-Р. 24-35.
6. Абрамов С. А. Задачи по программированию // С. А. Абрамов, Г. Г. Гнездилова, Е. Н. Капустина, М. И. Селюн/ — М.: Наука, 1988. – 256с.
7. Березин Б.И., Начальный курс Delphi // Б.И.Березин, С.Б.Березин / — М.: ДИАЛОГ-МИФИ, 1996. – 331с.
8. Бондарев В.М. Основы программирования // В.М. Бондарев, В.И. Рублинецкий, Е.Г. Качко / — Харьков: Фолио, Ростов н/Д: Феникс, 1997. – 446с.
9. Вирт Н. Алгоритмы и структуры данных. // Н. Вирт /— М.: Мир, 1989. - 345с.
- 10.Гладков В. П. Задачи по информатике на вступительном экзамене в вуз и их решения: Учебное пособие // В. П. Гладков / — Пермь: Перм. техн. ун-т, 1994. – 516с.
- 11.Грогоно П. Программирование на языке Delphi // П. Грогоно / —М.: Мир, 1982. – 216с.
- 12.Дагене В.А. 100 задач по программированию // В.А.Дагене, Г.К. Григас, К.Ф. Аугутис /— М.: Просвещение, 1993. – 106с.

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		58

13. Джамса К. Библиотека программиста Java // К. Джамса /- Jamsa Press, 1996, ООО "Попурри", 1996. – 656с.

14. Марков А. С. «Базы данных. Введение в теорию и методологию // А. С. Марков, К.Ю. Лисовский / - Финансы и статистика»-2006,-Р. 24-35.

15. Заварыкин В.М. Основы информатики и вычислительной техники // В.М. Заварыкин, В.Г. Житомирский, М.П. Лапчик / — М.: Просвещение, 1989. – 556с.

16. Касаткин В. Н. Информация. Алгоритмы. ЭВМ // В. Н. Касаткин / — М.: Просвещение, 1991. – 219с.

17. Кен А. Язык программирования Delphi // А. Кен, Дж. Гослинг /.- Addison-Wesley Longman, U.S.A., 1996, Издательство "Питер-Пресс", 1997. – 378с.

18. Керниган Б. Язык программирования Delphi // Б. Керниган, Д. Ритчи / Пер. с англ. — М.: Финансы и статистика, 1992. – 391с.

19. Ляхович В.Ф. Руководство к решению задач по основам информатики и вычислительной техники // В.Ф. Ляхович/ — М.: Высшая школа, 1994. – 127с.

20. Мейнджер Дж. Delphi Основы программирования // Дж. Мейнджер /- McGraw-Hill, Inc., 1996, Издательская группа BHV, Киев, 1997. – 346с.

21. Миков А. И. Информатика. Введение в компьютерные науки // А. И. Миков / — Пермь: Изд-во ПГУ, 1998. – 442с.

22. Могилев А. В. Информатика: Учеб. пособие для студ. пед. вузов // А. В. Могилев, Е. К. Хеннера./ — М.: Изд. центр «Академия», 1999. – 629с.

23. Нотон П. JAVA:Справ.руководство// П.Нотон, А.Тихонова.- М.:БИНОМ:Восточ.Кн.Компания,1996:Восточ.Кн.Компания. - 447с.

24. Нотон П. Полный справочник по Java // П. Нотон, Г. Шилдт//.- McGraw-Hill, 1997, Издательство "Диалектика", 1997. – 556с.

25. Ренеган Э.Дж. 1001 адрес WEB для программистов :Новейший путеводитель программиста по ресурсам World Wide Web:Пер.// Э.Дж. Ренеган /с англ..-Минск:Попурри,1997.-512с.ил.

26. Родли Дж. Создание Java-апплетов // Дж. Родли / - The Coriolis Group, Inc., 1996, Издательство НИПФ "ДиаСофт Лтд.", 1996. – 466с.

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

27. Секреты программирования для Internet на Java // М.Томас, П. Пратик, А. Хадсон, Д. Болл/ - Ventana Press, Ventana Communications Group, U.S.A., 1996, Издательство "Питер Пресс", 1997. – 396с.

28. Семакина И. Г. Информатика. Задачник-практикум: В 2 т. // И. Г. Семакина, Е.К.Хеннера/ — М.: Лаборатория Базовых Знаний, 1999. – 476с.

29. Сокольский М.В. Все об Intranet и Internet // М.В. Сокольский / - М.:Элиот, 1998.-254с.ил.

30. Тассел Д. Стил, разработка, эффективность, отладка и испытание программ // Д. Тассел / — М.: Мир, 1981. – 56с.

31. Тюрин Ю.Н. Анализ данных на компьютере./ Ю.Н. Тюрин, А.А. Макаров, В.Э.Фигурнова. - М.: ИНФРА-М, Финансы и статистика, 1995. - 384с.

32. Флэнэген Д. Java in a Nutshell / Д. Флэнэген /- O'Reilly & Associates, Inc., 1997, Издательская группа BHV, Киев, 1998. – 473с.

33. Чен М.С. Программирование на C++:1001 совет:Наиболее полное руководство по Java и Visual J++ :Пер.с англ.// М.С.Чен, С.В. Грифис, Э.Ф. Изи./ -Минск:Попурри,1997.-640с.ил.+ Прил.(1диск.).

34. Эферган М. C++: справочник // М. Эферган /.- QUE Corporation, 1997, Издательство "Питер Ком", 1998. – 256с.

35. Методичні рекомендації до виконання дипломного проекту з освітньо-кваліфікаційного рівня “Бакалавр” напряму підготовки 6.050102 «Комп’ютерна інженерія» фахового спрямування «Комп’ютерні системи та мережі» / О.М. Березький, Л.О.Дубчак, Р.Б. Трембач, Г.М. Мельник, Ю.М. Батько, С.В. Івасьєв / Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2014.–65 с.

36. Методичні вказівки до написання техніко-економічного розділу дипломних проектів освітньо-кваліфікаційного рівня «бакалавр» напряму підготовки 6.050102 комп’ютерна інженерія/ І.Р. Паздрій – Тернопіль: ТНЕУ, 2014. – 37 с.

					ДР.КСМ.110850/16.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		60