

# ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

## «Програмний додаток побудови кривих Гілберта»

Студентки 2м курсу, 2 групи,  
спеціальності 121 «Інженерія  
програмного забезпечення»  
освітньої програми «Інженерія  
програмного забезпечення»

Юрченка Станіслава  
Павловича

\_\_\_\_\_

підпис студента

Науковий керівник  
кандидат економічних наук,  
доцент кафедри інженерії  
програмного забезпечення та  
кібербезпеки

Тищенко Дмитро  
Олександрович

\_\_\_\_\_

підпис керівника

Гарант освітньої програми  
кандидат педагогічних наук,  
доцент кафедри інженерії  
програмного забезпечення та  
кібербезпеки

Котенко Наталія  
Олексіївна

\_\_\_\_\_

підпис гаранта

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь магістр

Освітня програма 121 «Інженерія програмного забезпечення»

### **Затверджую**

Зав. кафедри інженерії програмного  
забезпечення та кібербезпеки

Криворучко О. В.

«13» грудня 2022 р.

### **Завдання**

#### **на випускню кваліфікаційну роботу студентіві**

Юрченко Станіславу Павловичу

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи «Програмний додаток побудови  
кривих Гілберта»

Затверджена наказом ректора від «6» грудня 2022 р. № 3285

2. Строк здачі студентом закінченої роботи 27 листопада 2023

3. Цільова установка та вихідні дані до роботи

Мета роботи створення програмного додатку, який дозволить користувачам  
легко та ефективно побудовувати криві Гілберта різних порядків та  
використовувати їх у практичних задачах

Об'єкт дослідження є процес створення програмного додатку для побудови  
кривих Гілберта

Предмет дослідження є криві Гілберта



4. Консультанти роботи із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1 ТЕОРЕТИЧНА ОСНОВА

1.1. Огляд теорії кривих Гілберта

1.2. Математичні підходи до побудови кривих Гілберта

1.3. Застосування кривих Гілберта в науці та технологіях

1.4. Висновки до розділу 1

РОЗДІЛ 2 РОЗРОБКА ПРОГРАМНОГО ДОДАТКА

2.1. Вибір програмного середовища та мови програмування

2.2. Архітектура програмного додатка

2.3. Реалізація алгоритмів побудови кривих Гілберта

2.4. Візуалізація кривих Гілберта

2.5. Тестування та відладка програмного додатка

2.6. Висновки до розділу 2

РОЗДІЛ 3 ЕКСПЕРИМЕНТИ ТА РЕЗУЛЬТАТИ

3.1. Опис експериментального середовища

3.2. Тестові вхідні дані

3.3. Аналіз результатів експериментів

3.4. Порівняння реалізованих методів побудови кривих Гілберта

3.5. Визначення швидкості та точності програмного додатка

3.6. Висновок до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ПРОГРАМА ТА МЕТОДИКА ТЕСТУВАННЯ

ДОДАТКИ

## 6. Календарний план виконання роботи

№ пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускної кваліфікаційної роботи</i>	07.11.2022	07.11.2022
2.	<i>Розробка та затвердження завдання на роботу магістра (стац/заоч)</i>	13.12.2022	13.12.2022
3.	<i>Вступ та перелік літературних джерел</i>	24.02.2023	24.02.2023
4.	<i>Розробка технічного завдання</i>	15.03.2023	15.03.2023
5.	<i>Розділ 1. Теоретична основа</i>	10.04.2023	10.04.2023
6.	<i>Розділ 2. Розробка програмного додатка</i>	24.05.2023	24.05.2023
7.	<i>Розділ 3. Експерименти та результати</i>	06.09.2023	06.09.2023
8.	<i>Розробка програми та методики тестування</i>	18.10.2023	18.10.2023
9.	<i>Написання наукової статті</i>	17.05.2023	17.05.2023
10.	<i>Керівництво користувача</i>	25.10.2023	25.10.2023
11.	<i>Висновки та пропозиції</i>	01.11.2023	01.11.2023
12.	<i>Здача випускної кваліфікаційної роботи на кафедрі (перша перевірка)</i>	06.11.2023	06.11.2023
13.	<i>Підготовка автореферату та презентації доповіді</i>	06.11.2023	06.11.2023
14.	<i>Попередній захист випускної кваліфікаційної роботи</i>	20.11.2023 – 24.11.2023	20.11.2023 – 24.11.2023
15.	<i>Здача зброшурованої випускної кваліфікаційної роботи</i>	01.12.2023	01.12.2023
16.	<i>Зовнішнє рецензування випускної кваліфікаційної роботи</i>	02.12.2023	02.12.2023
17.	<i>Підготовка до публічного захисту випускної кваліфікаційної роботи</i>	05.12.2023- 06.12.2023	0 .12.2023-

7. Дата видачі завдання «13» грудня 2022 р.

8. Науковий керівник випускної кваліфікаційної роботи \_\_\_\_\_

Тищенко Д.О.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми \_\_\_\_\_

Котенко Н.О.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент \_\_\_\_\_

Юрченко С.П.

(прізвище, ініціали, підпис)





## АНОТАЦІЯ

Ця програмна робота присвячена розробці та реалізації алгоритмів для створення та візуалізації Кривих Гільберта на мові програмування Python. Криві Гільберта є особливими геометричними фігурами, які знаходять застосування в багатьох галузях, включаючи комп'ютерну графіку, топологію, архітектуру даних, інтерактивний дизайн та багато інших.

У цій роботі розглядаються теоретичні аспекти Кривих Гільберта, їх властивості та структура. Потім розробляються програмні алгоритми для генерації Кривих Гільберта різних порядків та їх візуалізації. Програмний код написаний на мові Python та використовує бібліотеки для роботи з графікою і візуалізації даних.

Результатом роботи є програмний додаток, який дозволяє користувачам генерувати та досліджувати Криві Гільберта різних порядків, а також відобразити їх на графічному інтерфейсі. Цей додаток може бути корисним для вивчення геометрії, візуалізації даних та розвитку навичок програмування на Python.

Ключові слова: Криві Гільберта, Python, візуалізація, графіка, геометрія, програмування.

Випускна кваліфікаційна робота на тему «Програмний додаток побудови кривих Гільберта» містить 54 сторінки, 8 рисунків і 3 таблиці. Перелік використаних джерел налічує 30 найменувань.



## ABSTRACT

This research project is dedicated to the development and implementation of algorithms for creating and visualizing Hilbert Curves in the Python programming language. Hilbert Curves are unique geometric figures with applications in various fields, including computer graphics, topology, data architecture, interactive design, and many others.

The work delves into the theoretical aspects of Hilbert Curves, their properties, and structure. Subsequently, software algorithms are developed for generating Hilbert Curves of different orders and visualizing them. The Python code is utilized, along with libraries for graphics and data visualization.

The outcome of this project is a software application that allows users to generate and explore Hilbert Curves of various orders, displaying them on a graphical interface. This application can be valuable for learning geometry, data visualization, and advancing Python programming skills.

**Keywords:** Hilbert Curves, Python, visualization, graphics, geometry, programming.

The graduation thesis on the topic " Software Application for Constructing Hilbert Curves" consists of 54 pages, 8 figures, and 3 tables. The list of references includes 30 sources.

"

<b>ЗМІСТ</b>	
<b>ВСТУП</b> .....	<b>3</b>
<b>РОЗДІЛ 1 ТЕОРЕТИЧНА ОСНОВА</b> .....	<b>7</b>
1.1. Огляд теорії кривих Гілберта .....	7
1.2. Математичні підходи до побудови кривих Гілберта .....	10
1.3. Застосування кривих Гілберта в науці та технологіях.....	13
1.4. Висновки до розділу 1 .....	17
<b>РОЗДІЛ 2 РОЗРОБКА ПРОГРАМНОГО ДОДАТКА</b> .....	<b>20</b>
2.1. Вибір програмного середовища та мови програмування .....	20
2.2. Архітектура програмного додатка .....	25
2.3. Реалізація алгоритмів побудови кривих Гілберта.....	30
2.4. Візуалізація кривих Гілберта.....	35
2.5. Тестування та відладка програмного додатка .....	41
2.6. Висновки до розділу 2.....	47
<b>РОЗДІЛ 3 ЕКСПЕРИМЕНТИ ТА РЕЗУЛЬТАТИ</b> .....	<b>52</b>
3.1. Опис експериментального середовища.....	52
3.2. Тестові вхідні дані .....	54
3.3. Аналіз результатів експериментів .....	55
3.4. Порівняння реалізованих методів побудови кривих Гілберта.....	56
3.5. Визначення швидкості та точності програмного додатка .....	57
3.6. Висновок до розділу 3 .....	58
<b>ВИСНОВКИ ТА ПРОПОЗИЦІЇ</b> .....	<b>60</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b> .....	<b>62</b>
<b>ТЕХНІЧНЕ ЗАВДАННЯ</b> .....	<b>65</b>
<b>ПРОГРАМА ТА МЕТОДИКА ТЕСТУВАННЯ</b> .....	<b>67</b>
<b>ДОДАТКИ</b> .....	<b>69</b>

					<i>ДТЕУ 121 02-25.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.		01.11.23	<i>Програмний додаток побудови кривих Гілберта</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник		Тищенко Д.О.		01.11.23		<i>Зміст</i>	2	64
Гарант		Котенко Н.О.		01.11.23		<i>Факультет інформаційних технологій 2м курс, 2 група</i>		
Розробив		Юрченко С.П.		01.11.23				
					<i>Зміст</i>			



## ВСТУП

У сучасному світі, відомому своєю безперервною та стрімкою еволюцією інформаційних технологій, існує постійна потреба в розробці нових математичних та алгоритмічних рішень для вирішення різноманітних завдань. Однією з цих завдань є побудова кривих Гільберта, які є важливим інструментом у різних галузях, таких як комп'ютерна графіка, геоінформаційні системи, архітектурне проектування та інші.

Криві Гільберта виникли в роботах шведського математика Давіда Гільберта у 1890 році та були вперше описані ним як фрактальні криві. Вони відзначаються своєю неперервністю та деталізованістю, що робить їх ідеальним інструментом для моделювання складних форм та об'єктів. Криві Гільберта мають також практичне застосування в областях, де необхідно розбивати простір на частини, зберігаючи при цьому гладкість та мінімізуючи величину обмежень на переміщення.

Однак, не зважаючи на довгу історію досліджень кривих Гільберта, існують ряд проблем та завдань, які досі залишаються нерозв'язаними або вимагають подальших досліджень. Новизна даного дипломного проекту полягає в розробці програмного додатку для побудови кривих Гільберта, який поєднує в собі не лише математичну точність, але й високу ефективність обчислень та зручний інтерфейс користувача. Цей додаток може стати інструментом, який допоможе вирішити багато практичних завдань, пов'язаних з генерацією та обробкою кривих Гільберта.

Актуальність теми обумовлена ростом інтересу до фрактальної геометрії, яка знаходить застосування в сучасних обчислювальних системах,

					<i>ДТЕУ 121 02-25.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>Програмний додаток побудови кривих Гілберта</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Зав. каф.	Криворучко О.В.			24.02.23		<i>В</i>	<i>3</i>	<i>64</i>
Керівник	Тищенко Д.О.			24.02.23		<i>Факультет інформаційних технологій</i>		
Гарант	Котенко Н.О.			24.02.23		<i>2м курс, 2 група</i>		
Розробив				24.02.23		<i>Вступ</i>		

медичній діагностиці, анімації та інших галузях. Вивчення кривих Гільберта дозволить розширити наше розуміння фрактальних об'єктів та їх можливостей, а також сприятиме подальшому розвитку обчислювальної геометрії

Метою даного дипломного проекту є створення програмного додатку, який дозволить користувачам легко та ефективно побудовувати криві Гільберта різних порядків та використовувати їх у практичних задачах. Для досягнення цієї мети ми будемо використовувати сучасні методи обчислення кривих Гільберта та розробляти зручний інтерфейс користувача, що дозволить навіть неспеціалістам легко працювати з цими фрактальними об'єктами.

У цьому дипломному проекті ми розглянемо математичну теорію кривих Гільберта, розробимо алгоритми їх побудови та створимо програмний додаток, який дозволить користувачам інтерактивно взаємодіяти з цими кривими. Такий додаток може знайти застосування в багатьох галузях, від дизайну інтерфейсів до наукових досліджень в галузі обчислювальної геометрії.

Дипломний проект також відкриває широкий простір для подальших досліджень у галузі фрактальної геометрії та обчислювальної геометрії. Вивчення та застосування кривих Гільберта в сучасних інформаційних технологіях може сприяти створенню нових методів оптимізації, комп'ютерної візуалізації, інтерактивного моделювання та багатьох інших сферах.

Сучасні вимоги до інформаційних систем та комп'ютерної графіки ставлять перед нами завдання забезпечити ефективну обробку складних об'єктів та їх візуалізацію. Криві Гільберта, завдяки своєму фрактальному характеру, можуть слугувати важливим компонентом вирішення цих завдань.

Даний проект розглядається в контексті загального розвитку обчислювальних методів та інструментів у галузі фрактальної геометрії та математичної моделювання. Розробка програмного додатку для побудови

						Аркуш
					ДТЕУ 121 02-25.МР	4
Зм.	Аркуш	№ докум	Підпис	Дата		



кривих Гільберта відкриває можливості для подальшого розвитку алгоритмів, оптимізації обчислень та створення нових застосувань у різних галузях науки та технологій.

Актуальність даного дипломного проекту підтверджується не лише теоретичними розв'язками, але і практичними можливостями застосування кривих Гільберта. Впровадження результатів цього дослідження в різні галузі може призвести до розширення можливостей інформаційних технологій, що відкриває нові горизонти для подальшого наукового та технологічного розвитку.

В рамках цього дипломного проекту ми розглядаємо також практичний аспект реалізації програмного додатку для побудови кривих Гільберта. Практичне використання цих кривих допомагає вирішити певні завдання у реальних проектах. Ми прагнемо надати інструмент, що дозволить використовувати криві Гільберта на практиці, спрощуючи завдання, які вимагають гладкої інтерполяції точок у просторі.

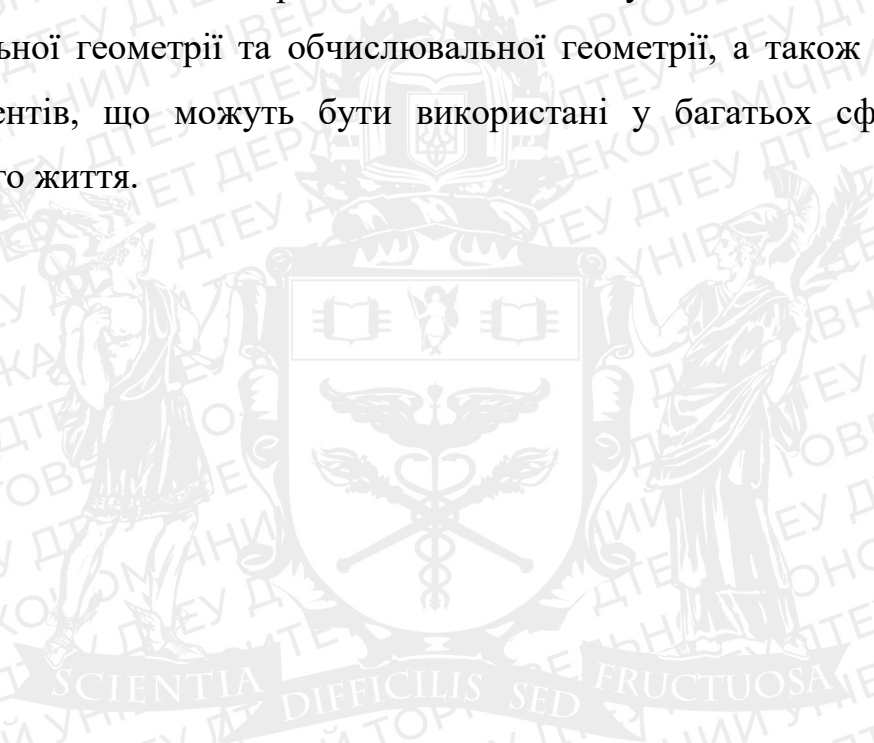
Важливість розробки програмного додатку для побудови кривих Гільберта стає зрозумілою при розгляді реальних прикладів використання. Наприклад, в комп'ютерній графіці криві Гільберта можуть використовуватися для створення складних рухомих шляхів для об'єктів, які надають реалістичну динаміку сценічним ефектам. В архітектурному проектуванні вони можуть бути використані для генерації складних форм будівель, що раціоналізує будівництво та покращує їх функціональність.

Також слід відзначити, що наукові дослідження в області фрактальної геометрії і фрактальних кривих важливі для розвитку алгоритмів комп'ютерного зору та обробки зображень. Використання кривих Гільберта може допомогти в розв'язанні завдань, пов'язаних із виявленням об'єктів на зображеннях та подальшою їх обробкою.

						Аркуш
					ДТЕУ 121 02-25.МР	5
Зм.	Аркуш	№ докум	Підпис	Дата		

Дипломний проект ініціює новий етап у вивченні та застосуванні кривих Гільберта, дозволяючи не лише розглядати їх теоретичні аспекти, але й створювати практичні рішення для широкого кола завдань. Ми сподіваємося, що ця робота стане початком для подальших досліджень та застосувань цих кривих у сучасному інформаційному середовищі.

Даний дипломний проект необхідний для удосконалення знань у галузі фрактальної геометрії та обчислювальної геометрії, а також для розвитку інструментів, що можуть бути використані у багатьох сферах нашого сучасного життя.



						ДТЕУ 121 02-25.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			6



## РОЗДІЛ 1

### ТЕОРЕТИЧНА ОСНОВА

#### 1.1. Огляд теорії кривих Гільберта

Криві Гільберта є цікавим математичним об'єктом, який отримав свою назву на честь шведського математика Давида Гільберта. Ці криві представляють собою фрактальні геометричні об'єкти, які мають безліч застосувань у різних галузях науки та інженерії.

Криві Гільберта відносяться до сімейства фрактальних кривих, які характеризуються тим, що вони є самоподібними на будь-якому масштабі. Основна ідея кривих Гільберта полягає в тому, що вони можуть бути побудовані шляхом ітеративного додавання менших сегментів до попередньої кривої, що дозволяє створювати складні фрактальні структури.

Сам процес побудови кривих Гільберта може бути визначений за допомогою рекурсивних правил. Починаючи з базової кривої, поділеної на декілька сегментів, на кожному кроці рекурсії кожен сегмент розділяється на менші сегменти, і вони замінюються на вже побудовані криві Гільберта меншого рівня складності[1]. Цей процес повторюється декілька разів, що призводить до збільшення деталізації та складності кривої.

Основні властивості кривих Гільберта включають їхню довжину, фрактальну розмірність та зв'язані з ними математичні властивості. Наприклад, криві Гільберта мають складності, яка зростає з кожним рівнем рекурсії, і їх фрактальна розмірність може бути обчислена за допомогою відповідних математичних формул.

Зм.	Аркуш	№ докум.	Підпис	Дата	ДТЕУ 121 02-25.МР			
Зав. каф.		Криворучко О.В.		10.04.23	Програмний додаток побудови кривих Гільберта	Стадія	Аркуш	Аркушів
Керівник		Тищенко Д.О.		10.04.23		Р1	7	64
Гарант		Котенко Н.О.		10.04.23	Теоретична основа	Факультет інформаційних технологій 2м курс, 2 група		
Розробив		Юрченко С.П.		10.04.23				

У світлі сучасних обчислювальних можливостей, побудова кривих Гільберта може бути використана в різних галузях, включаючи комп'ютерну графіку, комп'ютерне моделювання, геоінформатику та деякі аспекти штучного інтелекту. Дані криві також є важливими для досліджень в областях фрактальної геометрії та теорії хаосу.

Криві Гільберта демонструють фрактальну природу, що визначається їхньою самоподібністю та можливістю надзвичайно деталізованого розгляду при будь-якому масштабі. Ця фрактальність відображається в їхній розмірності, яка може бути нецілим числом. Чим більше рекурсій застосовується для побудови кривої Гільберта, тим більша буде її фрактальна розмірність, наближаючись до 2, що вказує на заповненість цієї кривої в площині (Рис. 1.1).

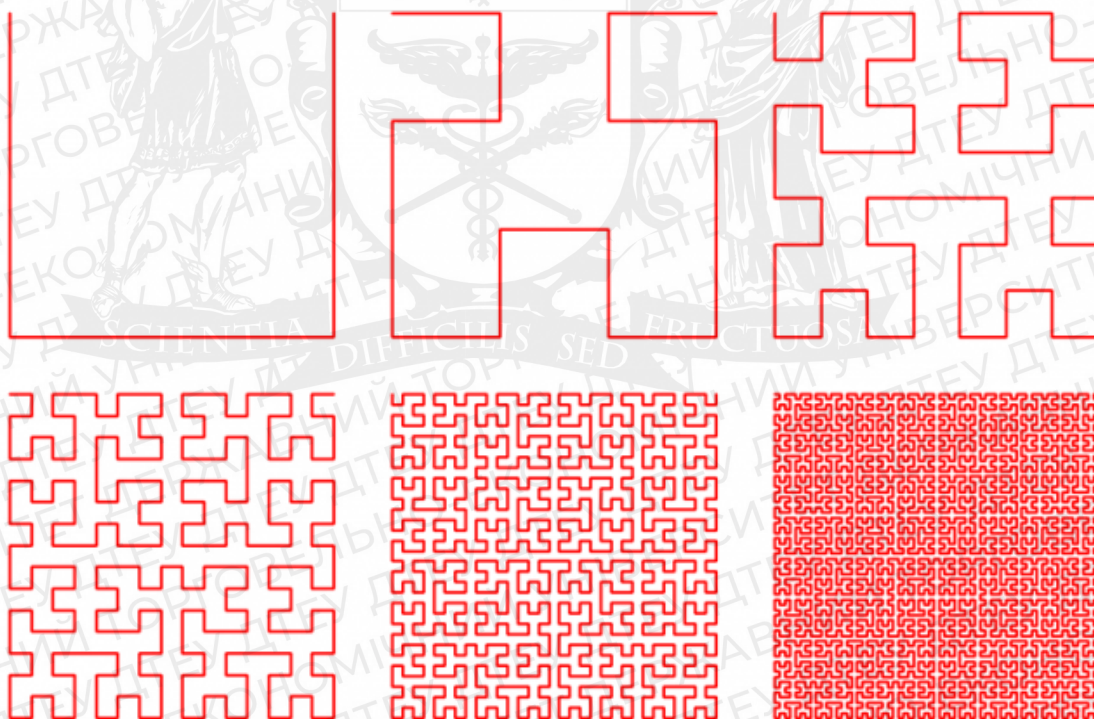


Рис. 1.1 – Загальний вигляд кривої Гільберта

Теорія кривих Гільберта також включає в себе питання їхньої довжини. Однією з центральних проблем є обчислення точної довжини кривих Гільберта. Навіть на вигляд простої кривої, вони розглядаються як складні математичні об'єкти, і їхню довжину важко знайти аналітично.

						ДТЕУ 121 02-25.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			8



Теорія кривих Гільберта також знаходить застосування у графічних системах і візуалізації даних. Побудова цих кривих може бути використана для створення складних геометричних об'єктів на комп'ютерних екранах, що дозволяє реалізувати відмінні візуальні ефекти та створювати складні шляхи для об'єктів у графіці[2].

Нарешті, важливо відзначити, що криві Гільберта не є обмеженими двовимірними структурами. Теорія розширюється на вищі розмірності, що означає, що криві Гільберта можуть бути побудовані у тривимірному, чотириразовому та вищих просторах.

Поняття кривих Гільберта також відіграють важливу роль у геометрії та топології. Вони представляють собою приклади фрактальних кривих, які допомагають вивчати особливості фрактальних об'єктів і розвивати методи їхнього аналізу. Такі властивості, як самоподібність та неперервність на всьому своєму діапазоні масштабів, роблять криві Гільберта цікавими об'єктами для дослідження.

Застосування кривих Гільберта можна знайти у візуальному мистецтві та дизайні. Вони можуть бути використані для створення художніх інсталяцій, комп'ютерних гравюр і абстрактних образів. Ця фрактальна геометрія надає можливість творити візуально захоплюючі та складні образи, які стають джерелом натхнення для багатьох митців.

Крім того, криві Гільберта мають своє місце у теорії даних та обробці сигналів. Вони можуть бути використані для створення зшивок, які допомагають у покращенні якості зображень та відео, а також у стисненні даних. Вивчення їхньої структури і властивостей сприяє розвитку нових алгоритмів обробки сигналів та зменшенню обсягу даних.

На підсумок, теорія кривих Гільберта є важливою галуззю математики та має різнобічні застосування у багатьох сферах. Розуміння їхньої природи та властивостей відкриває широкий спектр можливостей для теоретичних

						Аркуш
					ДТЕУ 121 02-25.МР	9
Зм.	Аркуш	№ докум	Підпис	Дата		

досліджень та практичних застосувань у науці та технологіях. Побудова програмного додатку для побудови кривих Гільберта може бути стимулом для подальших досліджень та розвитку нових інновацій.

## 1.2. Математичні підходи до побудови кривих Гільберта

Криві Гільберта є важливою математичною конструкцією, яка виникла в теорії фракталів та комп'ютерній графіці. Ці криві мають низьку фрактальну розмірність і відомі своєю здатністю заповнювати площину без розривів. У цьому розділі ми розглянемо основні математичні підходи до побудови кривих Гільберта та детально проаналізуємо їхню структуру та властивості.

Перш за все, варто зазначити, що криві Гільберта базуються на концепції самоподібності та рекурсії. Ця концепція полягає в тому, що крива подібна до самої себе на різних масштабах, і для її побудови використовується рекурсивний процес. Найпростіший спосіб побудувати криву Гільберта - це взяти лінію та розбити її на чотири рівні відрізки, після чого об'єднати їх в специфічному порядку, створюючи більш складну структуру. Цей процес може бути повторений багато разів, і кожен новий рівень додає більше деталей та складності кривій[3].

Одним із важливих математичних понять, пов'язаних з кривими Гільберта, є фрактальна розмірність. Це числова характеристика, яка вимірює складність фрактала. Криві Гільберта відзначаються низькою фрактальною розмірністю, що означає, що вони мають дуже складну форму, але при цьому займають мало простору. Це робить їх цікавими для застосувань у комп'ютерній графіці, де ефективне заповнення областей є важливою задачею.

Одним з математичних підходів до побудови кривих Гільберта є використання фрактальних L-systems. L-системи - це формальні граматики, які використовуються для генерації фрактальних структур. Вони визначають

						ДТЕУ 121 02-25.МР	Аркуш
							10
Зм.	Аркуш	№ докум	Підпис	Дата			



правила заміни символів в послідовності та можуть бути використані для створення рекурсивної послідовності кроків для побудови кривої Гілберта.

Ще одним підходом є використання математичних рівнянь, таких як крива Пеано, яка також є самоподібною і має схожі властивості з кривими Гілберта. Рівняння Пеано дозволяють точно визначити координати кожного вузла кривої, що робить їх корисними для точного моделювання та аналізу.

Задачі, пов'язані з кривими Гілберта, виникають у багатьох галузях науки і техніки. Однією з основних областей застосування є комп'ютерна графіка. Криві Гільберта використовуються для заповнення областей на екрані, створення складних графічних об'єктів та анімації. Їх фрактальна структура дозволяє створювати деталізовані зображення з мінімальними обчислювальними ресурсами. Щоб провести короткий огляд математичних підходів до побудови кривих Гілберта та їхніх застосувань у різних галузях науки та техніки було створено таблицю 1.1.

Таблиця 1.1

### Підходи до побудови кривих Гілберта та їх застосування

Підхід	Опис
Самоподібність та рекурсія	Основний принцип побудови кривих Гілберта полягає в рекурсивному діленні лінії на частини та їх подальшому з'єднанні в певному порядку. Цей процес ґрунтується на самоподібності структури кривої.
Фрактальна розмірність	Фрактальна розмірність - це числова характеристика, що вимірює складність фрактала. Криві Гілберта відзначаються низькою фрактальною розмірністю, що дозволяє їм ефективно заповнювати площину.
L-системи	L-системи - це формальні граматики, використовувані для генерації фрактальних структур. Вони визначають правила заміни символів у послідовності, що можуть бути використані для побудови кривої Гілберта.
Математичні рівняння	Криві Гілберта можуть бути описані математичними рівняннями, такими як крива Пеано. Ці рівняння дозволяють точно визначити координати кожного вузла кривої, що корисно для точного моделювання та аналізу.

Використання фрактальних властивостей	Фрактальні властивості кривих Гілберта роблять їх корисними у комп'ютерній графіці для заповнення областей, створення складних графічних об'єктів та анімації.
Застосування у теорії комп'ютерних наук	Криві Гілберта використовуються для побудови послідовностей, які мають особливості, корисні для обробки та пошуку даних. Наприклад, вони можуть бути використані для структурування даних у базах даних.
Геоінформатика та геодезія	Криві Гілберта можуть бути використані для моделювання географічних маршрутів та розрахунку оптимальних шляхів.
Застосування у телекомунікаціях	Криві Гілберта можуть бути використані для побудови антен та електронних пристроїв зі складними геометричними формами, що підвищує їхню функціональність та продуктивність.

У теорії комп'ютерних наук, криві Гільберта використовуються для побудови послідовностей, які мають особливості, корисні для обробки та пошуку даних. Наприклад, вони можуть бути використані для структурування даних у базах даних або для створення алгоритмів сортування.

У геоінформатиці та геодезії, криві Гільберта можуть бути використані для моделювання складних географічних маршрутів або для розрахунку оптимальних шляхів.

Крім того, криві Гільберта мають важливе застосування у телекомунікаціях та електроніці. Вони можуть бути використані для побудови антен та інших електронних пристроїв зі складними геометричними формами, що допомагає підвищити їхню функціональність та продуктивність.

Заключно, математичні підходи до побудови кривих Гільберта відіграють важливу роль у різних галузях науки та техніки, зокрема в комп'ютерній графіці, теорії комп'ютерних наук, геоінформатиці, геодезії та телекомунікаціях[4]. Розуміння їхньої математичної природи та властивостей є важливим для розвитку сучасних технологій та наукових досліджень. У

						Аркуш
						12
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-25.МР	



наступних розділах дипломного проекту буде розглянуто конкретні застосування кривих Гільберта в цих галузях та подальший розвиток цієї теми.

### 1.3. Застосування кривих Гільберта в науці та технологіях

Криві Гільберта, як складні фрактальні об'єкти, знайшли широке застосування в науці та технологіях. Їх структурна складність та унікальні властивості роблять їх корисними для різних областей. У цьому розділі ми розглянемо основні аспекти застосування кривих Гільберта у наукових дослідженнях та сучасних технологіях.

Одним із найважливіших використань кривих Гільберта є їх роль в геометричних моделях складних природних об'єктів. Наприклад, в географії криві Гільберта використовуються для подання границь земельних ділянок та контурів складних ландшафтів. Вони також застосовуються в біології для аналізу складних геномів та молекулярних структур.

Іншим важливим аспектом є використання кривих Гільберта в комп'ютерних науках і інформатиці. Вони широко використовуються для компактного збереження та передачі даних. Криві Гільберта можуть ефективно компресувати зображення, відео та географічні дані, зменшуючи обсяг інформації, не втрачаючи якості.

Застосування кривих Гільберта також знаходиться в сфері телекомунікацій, де вони допомагають у побудові оптимальних шляхів для передачі сигналів. Використання цих кривих допомагає зменшити затримки та підвищити ефективність мереж[5].

У фізиці криві Гільберта використовуються для аналізу фрактальних структур та об'єктів, таких як розгалужені річкові системи та фізичні фрактали. Вони допомагають розуміти та моделювати складні фізичні явища, що базуються на фрактальних принципах.

						ДТЕУ 121 02-25.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			13

У сучасних технологіях, таких як штучний інтелект та машинне навчання, криві Гільберта можуть бути використані для структурування та аналізу даних. Вони дозволяють побудувати багатосарові моделі, які відображають ієрархічну природу даних, сприяючи точнішому прогнозуванню та класифікації.

Додатково, криві Гільберта використовуються в географічних інформаційних системах (ГІС). Вони дозволяють ефективно подавати географічні об'єкти і регіони, забезпечуючи зручний спосіб аналізу геопросторових даних та навігації. Криві Гільберта можуть бути використані для побудови ієрархічних структур карт, що спрощує взаємодію з географічною інформацією та сприяє її збереженню та передачі.

У сфері архітектури та дизайну, криві Гільберта використовуються для створення креативних та нестандартних форм та структур. Вони можуть бути в основі архітектурних проєктів, які надають будівлям та спорудам унікальний вигляд. Криві Гільберта дають архітекторам і дизайнерам можливість експериментувати зі складними геометричними формами та створювати інноваційні рішення.

Ще одним застосуванням кривих Гільберта є в області медицини та біоінформатики. Вони використовуються для аналізу структури ДНК, моделювання молекулярних процесів та для досліджень в галузі нейробіології. Криві Гільберта допомагають розкрити складність біологічних систем та процесів, що має важливе значення для розвитку нових методів діагностики та лікування.

Навіть у мистецтві та розвагах криві Гільберта знаходять своє місце. Вони можуть бути використані для створення візуально захопливих мистецьких інсталяцій та комп'ютерних ігор[6]. Їх складні форми і структури надають можливість творити цікаві візуальні та аудіовізуальні ефекти, які захоплюють глядачів і гравців.

						ДТЕУ 121 02-25.МР	Аркуш
							14
Зм.	Аркуш	№ докум	Підпис	Дата			



Крім того, криві Гільберта відіграють важливу роль у геодезії та картографії. Вони використовуються для подання складних географічних об'єктів, таких як кордони країн, географічні характеристики природних резерватів та дорожні мережі. Криві Гільберта допомагають створювати картографічні продукти, які забезпечують точне та зручне відображення географічних об'єктів.

У сучасних технологіях, таких як віртуальна реальність і розширена реальність, криві Гільберта можуть бути використані для створення складних інтерактивних ігор і симуляцій. Вони дозволяють створити середовища, які відображають взаємодію користувача з віртуальним світом і можуть бути використані в освіті, розвагах та тренуваннях.

Криві Гільберта також грають важливу роль у технічних науках та інженерії. Вони можуть бути використані для побудови оптимальних шляхів для рухомих об'єктів, таких як роботи та автономні автомобілі. Використання кривих Гільберта допомагає покращити автономію та ефективність транспортних систем.

Завдяки своїй унікальній структурі, криві Гільберта також використовуються у криптографії для створення складних шифрів та методів обчислення. Вони забезпечують безпеку обміну інформацією та дозволяють створювати алгоритми, які важко розкрити.

Окрім того, криві Гільберта використовуються в геології та дослідженнях земної кори. Вони можуть бути використані для аналізу складних геологічних формацій, які мають фрактальну структуру. Це допомагає геологам та геофізикам краще розуміти геологічні процеси та прогнозувати землетруси та інші природні катастрофи.

Криві Гільберта знаходять застосування і в економічних дослідженнях. Вони можуть бути використані для аналізу складних фінансових ринків та

						Аркуш
						15
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-25.МР	

моделювання ієрархічних економічних систем. Це допомагає в управлінні ризиками та прийнятті рішень у сфері фінансів та економіки.

У сучасному медичному дослідженні, криві Гільберта можуть бути використані для аналізу складних медичних даних, таких як генетичні послідовності та зображення мозку. Вони допомагають вивчати геометричну структуру біологічних об'єктів та виявляти закономірності у хворобах і тах надають можливість розробляти індивідуальні методи лікування.

Науковці також використовують криві Гільберта для досліджень у сфері штучного життя та еволюційних моделей. Вони можуть бути використані для моделювання розвитку живих систем та вивчення процесів адаптації та еволюції.

Наступна таблиця 1.2 надає загальний огляд застосувань кривих Гільберта в науці та технологіях.

Таблиця 1.2

### Застосування кривих Гільберта в науці та технологіях

Галузь	Застосування кривих Гільберта
Географія	- Визначення географічних меж і контурів ландшафтів
Біологія	- Аналіз геномів та біологічних молекул
Комп'ютерні науки	- Компресія та збереження даних
Телекомунікації	- Оптимізація шляхів передачі сигналів
Фізика	- Моделювання фрактальних структур
Штучний інтелект	- Структурування та аналіз даних
Геодезія та картографія	- Подання географічних об'єктів на мапах
Архітектура і дизайн	- Створення нестандартних форм та структур
Медицина і біоінформатика	- Аналіз структури ДНК та молекулярних процесів
Мистецтво та розваги	- Створення візуально захопливих ефектів
Геологія	- Аналіз геологічних формацій і процесів
Економічні науки	- Аналіз фінансових ринків та економічних систем
Медичні дослідження	- Аналіз медичних даних та генетичних послідовностей
Штучний інтелект	- Моделювання розвитку живих систем



Криптографія

- Створення складних шифрів та методів обчислення

Узагальнюючи, криві Гільберта виявляються надзвичайно корисними в різних галузях науки та технологій завдяки своїй геометричній складності та унікальним властивостям. Вони служать як інструмент для аналізу та моделювання складних структур, так і для створення нових технологій та інновацій. Важливість кривих Гільберта в наукових дослідженнях та сучасних технологіях постійно росте, розширюючи їхню роль у сучасному світі. Роль кривих Гільберта у сучасному світі практично неоціненна і продовжує зростати з розвитком науки і технологій.

#### 1.4. Висновки до розділу 1

Криві Гільберта, які також відомі як фрактальні криві, є важливим елементом в сучасній математиці та інформатиці. Ці криві, розроблені шведським математиком Давідом Гільбертом у 1890 році, є прикладом самоподібних структур, що відображають багато важливих математичних та обчислювальних властивостей. Теорія, що стоїть за цими кривими, вивчається в різних галузях науки, включаючи теорію обчислень, геометрію, теорію чисел та інші.

Криві Гільберта відзначаються своєю самоподібністю, що означає, що кожен їх елемент має схожу структуру з більшими частинами. Ця властивість робить їх важливими в контексті комп'ютерної графіки та візуалізації, оскільки вони можуть бути використані для створення складних та деталізованих зображень. Однак побудова кривих Гільберта вимагає спеціалізованих алгоритмів та математичних інструментів.

У теоретичному аспекті, для розуміння кривих Гільберта необхідно вивчити поняття рекурсивних функцій та теорії фракталів. Рекурсивні

						ДТЕУ 121 02-25.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			17

функції грають важливу роль у побудові цих кривих, оскільки кожен їхній елемент обчислюється шляхом повторення певної операції на менших частинах кривої. Це дозволяє створити криву Гільберта будуючи її з менших підрозділів, що робить її структуру самоподібною.

З точки зору фракталів, криві Гільберта є класичним прикладом. Вони демонструють властивість самоподібності, яка робить їх цікавими об'єктами вивчення. Ця самоподібність може бути описана математично та використана для створення різних образів та зображень, від абстрактних фракталів до складних патернів у природі.

Далі важливо відзначити, що криві Гільберта не обмежуються тільки математичними абстракціями. Вони мають конкретні практичні застосування в інформатиці і комп'ютерній графіці. Наприклад, ці криві використовуються в розробці антенних систем для побудови антенних ліній, де їх самоподібна природа дозволяє покращити роботу антени на різних частотах. Вони також знаходять застосування в алгоритмах для стискання зображень та відео, що дозволяє ефективно зберігати та передавати великі об'єми інформації.

Важливим аспектом теоретичної основи є також вивчення алгоритмів побудови кривих Гільберта. Розробка алгоритмів для генерації цих кривих є важливою задачею у світі інформатики. Ці алгоритми вимагають глибокого розуміння теоретичних аспектів кривих Гільберта та включають в себе математичні обчислення, які дають змогу побудувати ці фрактальні структури з високою точністю та ефективністю.

Відзначимо, що криві Гільберта також мають значення в освіті та популяризації науки. Вони можуть бути використані як візуальні приклади для введення студентів та загальної громадськості в світ математики та фракталів. Такі візуальні засоби можуть стимулювати інтерес до науки та математики та сприяти їхньому вивченню.

						ДТЕУ 121 02-25.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			18



Теоретична основа побудови кривих Гільберта також глибоко пов'язана з поняттями комп'ютерних алгоритмів та обчислювальної геометрії. Розвиток алгоритмів для генерації цих кривих сприяє не лише створенню самоподібних структур, але і розробці більш загальних методів обробки графічних даних. Такі алгоритми можуть бути використані у різних застосунках, від комп'ютерних ігор до візуалізації наукових даних.

Додатково, теоретична основа кривих Гільберта вносить вклад у розвиток геометричних та топологічних аспектів математики. Самоподібність кривих Гільберта викликає питання про їх фрактальну природу та відкриває нові можливості для дослідження топологічних властивостей, наприклад, їх вимірності та складності.

Крім того, теоретична основа також пов'язана з іншими областями науки, такими як теорія обчислень, аналіз алгоритмів та дискретна математика. Побудова кривих Гільберта може включати в себе розгляд чисел Фрейгута, що є важливими для зрозуміння властивостей цих кривих.

У підсумку, теоретична основа побудови кривих Гільберта включає в себе розуміння рекурсивних функцій, теорії фракталів та їх властивостей самоподібності. Ця теорія відіграє важливу роль у сучасних обчислювальних науках та комп'ютерній графіці, де криві Гільберта використовуються для створення складних і креативних зображень. Розуміння цих аспектів дозволяє розширити уявлення про математичні структури та їхні застосування у сучасному світі. Криві Гільберта, завдяки своїм фундаментальним властивостям, залишаються об'єктом активного дослідження та застосування в різних галузях науки та технології. Дослідження цієї теорії дозволило розширити наше розуміння самоподібних структур та їх застосувань у різних галузях науки і технологій.

						Аркуш
					ДТЕУ 121 02-25.МР	19
Зм.	Аркуш	№ докум	Підпис	Дата		

## РОЗДІЛ 2

### РОЗРОБКА ПРОГРАМНОГО ДОДАТКА

#### 2.1. Вибір програмного середовища та мови програмування

Вибір програмного середовища та мови програмування є критичним етапом в розробці програмного додатку для побудови кривих Гільберта. В цьому розділі розглянемо різні аспекти, які обумовлюють вибір Python як мови програмування та вибір програмного середовища для розробки даного додатку.

##### *Мова програмування Python*

Python обрана як мова програмування для реалізації програмного додатку з декількох важливих причин. По-перше, Python відома своєю простотою і легкістю вивчення, що робить її доступною для розробників з різним рівнем досвіду. Ця мова також має велику та активну спільноту розробників, що веде до постійного розвитку та підтримки.

По-друге, Python має багатий список бібліотек і фреймворків, які допомагають зробити розробку програмного додатку більш ефективною та швидкою[7]. Для побудови кривих Гільберта, необхідні математичні обчислення, і Python має багато бібліотек, таких як NumPy та SciPy, які полегшують цей процес.

По-третє, Python підтримує крос-платформеність, що означає, що програмний додаток може бути запущений на різних операційних системах без великих зусиль. Це важливо для того, щоб зробити додаток доступним для широкого кола користувачів.

##### *Програмне середовище для розробки*

					<i>ДТЕУ 121 02-25.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.		24.05.23	<i>Програмний додаток побудови кривих Гілберта</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник		Тищенко Д.О.		24.05.23		P2	20	64
Гарант		Котенко Н.О.		24.05.23		<i>Факультет інформаційних технологій 2м курс, 2 група</i>		
Розробив		Юрченко С.П.		24.05.23				
					<i>Розробка програмного додатку</i>			



Для розробки програмного додатку для побудови кривих Гільберта, вибір програмного середовища також є важливим аспектом. Однак в даному випадку можливо вибрати з різних програмних середовищ, з урахуванням специфіки завдання.

Щоб забезпечити зручну і продуктивну роботу, можна використовувати інтегровані середовища розробки (IDE), такі як PyCharm, Visual Studio Code або Jupyter Notebook. Ці IDE надають різні зручності, такі як автодоповнення, налагодження коду та візуалізація результатів, що полегшують розробку та налагодження.

З іншого боку, для простих або експериментальних завдань можна використовувати інтерактивне середовище, таке як Jupyter Notebook. Воно дозволяє виконувати код по частинах та надавати пояснення до кожного кроку, що допомагає у розумінні алгоритму побудови кривих Гільберта.

#### *Середовище Visual Studio Code (VS Code) для програмування на Python*

Visual Studio Code - це популярне і легкове інтегроване середовище розробки, розроблене компанією Microsoft. VS Code відкритий, розширюваний та має велику активну спільноту розробників. Для роботи з Python в VS Code доступні різні розширення та налаштування, які роблять його потужним інструментом для розробки на цій мові (Рис.2.1).

						ДТЕУ 121 02-25.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			21

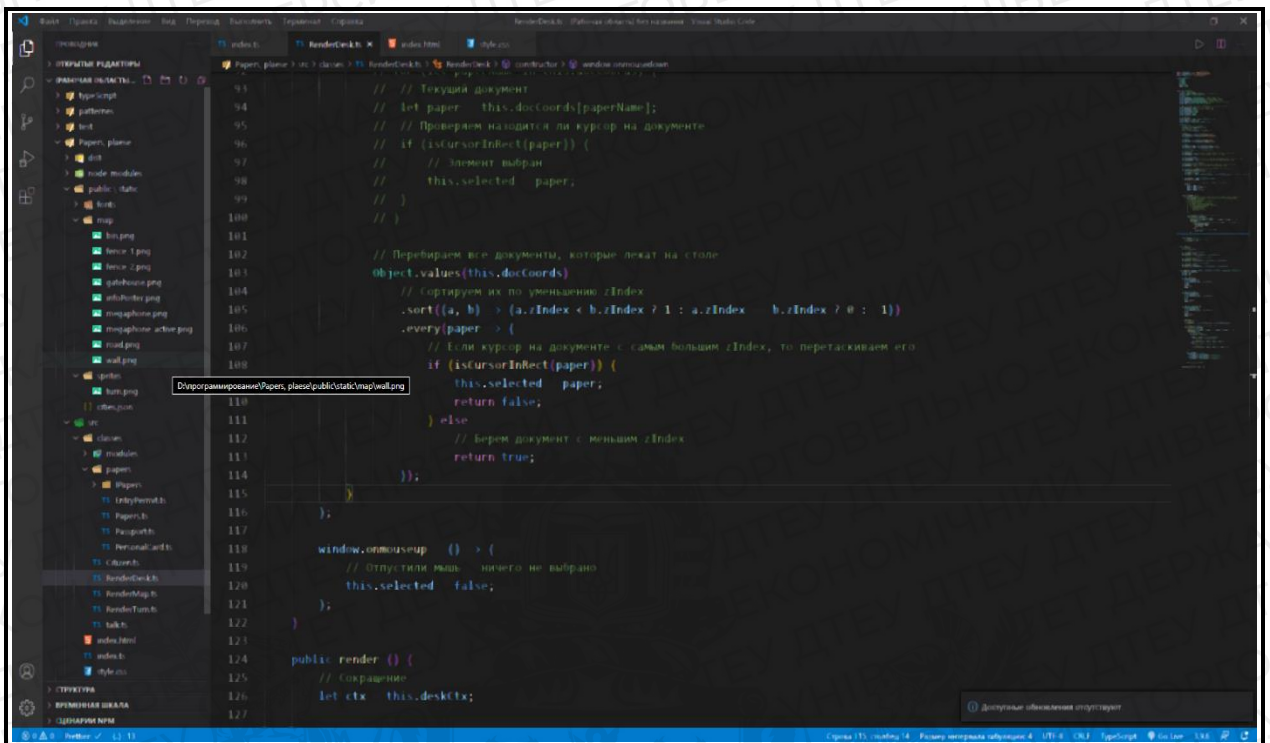


Рис. 2.1 – Вигляд робочого вікна програми

### *Встановлення Python у VS Code*

У VS Code існують різні розширення для роботи з Python. Один з найпопулярніших пакунків - це "Python" від Microsoft, який дозволяє легко встановити та налаштувати робоче середовище Python. Це розширення надає такі можливості, як автодоповнення, відстеження помилок та можливість відладки.

У VS Code робота відбувається в "робочому просторі" (workspace), який представляє собою проект або папку з вашими файлами Python. Можна відкрити свій робочий простір, створити новий файл або відредагувати існуючі файли.

VS Code дозволяє легко налаштувати робочий процес. Можна налаштувати використання конкретної версії Python, встановити віртуальне середовище, і налаштувати інші параметри через файл `settings.json`. Крім того, VS Code має вбудовану підтримку для відлагодження, що дозволяє вам легко відстежувати та виправляти помилки у вашому коді.

						Аркуш
						22
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-25.МР	



VS Code - це дуже розширюване середовище, і можна встановлювати різні розширення для полегшення роботи з Python. Наприклад, можна встановити розширення для роботи з системами контролю версій, або для роботи з базами даних, якщо вони вам потрібні у вашому проєкті.

VS Code має зручний текстовий редактор з багатьма корисними функціями, такими як автоматична перевірка синтаксису, автодоповнення, підсвічування синтаксису та багато інших[8]. Це допомагає писати та редагувати код ефективно та зручно.

Visual Studio Code є потужним середовищем розробки для Python. Воно надає зручний інтерфейс, багато корисних розширень та налаштувань, які роблять роботу з Python легкою та продуктивною. Вибір VS Code для програмування на Python є важливим кроком у створенні програмного додатку для побудови кривих Гільберта.

#### *Особливості та переваги програмування на Python*

Python - це мова програмування високого рівня, яка здобула велику популярність завдяки своїм особливостям і перевагам. Ось деякі з них:

1. Простота і читабельність коду: Python відомий своєю чистою та легко читабельною синтаксичною структурою. Це робить його ідеальним вибором для початківців та для проєктів, де розробники мають різний рівень досвіду.
2. Багатофункціональність: Python має велику кількість стандартних бібліотек і модулів, які полегшують різні завдання, включаючи обробку рядків, роботу з мережами, обчислення і багато інших.
3. Велика спільнота розробників: Python має активну та велику спільноту розробників, що означає, що завжди є доступ до допомоги, порад та різноманітних розширень.

						Аркуш
					ДТЕУ 121 02-25.МР	23
Зм.	Аркуш	№ докум	Підпис	Дата		

4. Переносимість: Python є крос-платформеним, що означає, що код, написаний на одній платформі, може легко переноситися на інші. Це робить його ідеальним для створення крос-платформених додатків.

5. Інтерпретована мова: Python - це інтерпретована мова, що означає, що вам не потрібно компілювати код перед його виконанням. Це спрощує і швидше розробку.

Підтримка об'єктно-орієнтованого програмування (ООП): Python підтримує ООП, що дозволяє створювати більш структуровані та повторно використовувані кодові блоки.

Можливість взаємодії з іншими мовами: Python легко інтегрується з іншими мовами програмування, такими як C, C++ і Java, що робить його універсальним для різних завдань.

Багато різноманітних бібліотек та фреймворків: Python має широкий вибір бібліотек і фреймворків для різних галузей розробки, включаючи веб-розробку (наприклад, Django і Flask), наукові обчислення (NumPy і SciPy), штучний інтелект (TensorFlow і PyTorch), обробку даних (Pandas) і багато інших[9].

Активна підтримка і розвиток: Python постійно розвивається і оновлюється, що робить його актуальним для сучасних завдань розробки.

Загалом, Python - це потужна і проста мова програмування, яка підходить для широкого кола завдань. Його особливості та переваги роблять його важливим інструментом для розробників у багатьох галузях програмування.

У кінцевому підсумку, вибір мови програмування Python та програмного середовища залежить від конкретних потреб та досвіду розробника. У даному дипломному проєкті, було обрано Python як мову програмування через його загальну придатність та багатий інструментарій для математичних обчислень. Щодо програмного середовища, можливо використовувати інтегроване

						ДТЕУ 121 02-25.МР	Аркуш
							24
Зм.	Аркуш	№ докум	Підпис	Дата			



середовище розробки, таке як PyCharm, для забезпечення комфортної та продуктивної роботи

## 2.2. Архітектура програмного додатка

Архітектура програмного додатка для побудови кривих Гільберта виконує центральну роль у забезпеченні ефективного та коректного функціонування програми. У цьому розділі ми розглянемо всі ключові аспекти цієї архітектури, включаючи структуру програми, використані бібліотеки, алгоритми та методи.

### *Структура програми*

Програма починається з імпорту необхідних бібліотек для малювання та створення графічного інтерфейсу. Потім визначаються функції для малювання кривої Гільберта та управління програмою. Головна функція для малювання рекурсивно визначає форму кривої Гільберта, використовуючи бібліотеку turtle[10]. Для малювання кривої з введеними користувачем параметрами викликається відповідна функція.

Після цього створюється графічний інтерфейс з полями для введення параметрів та кнопками для малювання, повернення до введення параметрів та виходу з програми.

Завершується програма запуском головного циклу обробки подій, який дозволяє користувачам взаємодіяти з програмою через графічний інтерфейс.

### *Алгоритми та методи*

Основним алгоритмом для побудови кривих Гільберта є рекурсивний підхід, який ґрунтується на математичних відношеннях та рекурсивних викликах функцій. Для обчислення точок на кривій Гільберта використовуються методи тригонометричних обчислень та геометричних перетворень.

						ДТЕУ 121 02-25.МР	Аркуш
							25
Зм.	Аркуш	№ докум	Підпис	Дата			

Окрім основного алгоритму для побудови кривих Гілберта, програма містить додаткові функції для обчислення довжини кривих, розрахунку координат точок, збереження результатів у файловій системі та відображення графічного представлення кривих за допомогою бібліотеки Matplotlib.

Підсистема візуалізації графіки в програмі реалізована з використанням об'єктно-орієнтованого підходу, що дозволяє створювати графічні об'єкти та взаємодіяти з ними, забезпечуючи високу гнучкість та легку розширюваність відображення результатів побудови кривих Гілберта[11].

Загальна архітектура програмного додатка для побудови кривих Гілберта розроблена з урахуванням принципів модульності, читабельності та ефективності, що дозволяє легко розширювати функціональність та вдосконалювати додаток у майбутньому.

Додатково, важливим аспектом архітектури програмного додатка є підтримка параметризації та налаштувань. Програма дозволяє користувачу задавати різні параметри побудови кривих Гілберта, такі як рівень рекурсії, довжину кривої, тип та розмір зображення тощо. Це робить програму більш універсальною та дозволяє використовувати її для створення різних видів кривих Гілберта з різними характеристиками.

Паралельна обробка та оптимізація є ще одним важливим аспектом архітектури програми. Для побудови кривих Гілберта на великій кількості рівнів рекурсії може виникнути проблема продуктивності[12]. Тому програма використовує механізми паралельної обробки, дозволяючи використовувати багатоядерні процесори для прискорення обчислень.

Забезпечення стабільності та коректності роботи програми є важливим завданням архітектури. Додаток включає в себе механізми обробки помилок, валідації введених даних та відслідковування стану виконання програми, що допомагає уникнути непередбачуваних ситуацій та забезпечує стабільну роботу.

						Аркуш
					ДТЕУ 121 02-25.МР	26
Зм.	Аркуш	№ докум	Підпис	Дата		



Усе вищезазначене створює потужну та гнучку архітектуру програмного додатка для побудови кривих Гілберта. Вона дозволяє користувачам з легкістю використовувати програму для створення та аналізу кривих Гілберта, а також надає можливість розширювати функціональність та оптимізувати продуктивність програми.

Додатково, важливим компонентом архітектури програмного додатка є система керування пам'яттю. Рекурсивний алгоритм побудови кривих Гілберта може призвести до великого обсягу даних та використання пам'яті, і ця проблема вимагає уважної оптимізації та контролю. Програма використовує механізми кешування результатів обчислень, щоб уникнути зайвого перерахунку та зекономити ресурси пам'яті.

Архітектура програмного додатка також передбачає можливість розширення функціональності шляхом додавання нових алгоритмів для побудови різних видів кривих, а також можливість інтеграції з іншими програмами чи бібліотеками. Додаток розроблений з урахуванням принципів модульності та інтерфейсів, що полегшує розширення та співпрацю з іншими розробниками.

Іншим важливим аспектом архітектури є безпека даних та взаємодія з користувачем. Програма враховує можливі ризики, пов'язані з введенням даних користувачем, та використовує механізми фільтрації та валідації для запобігання атакам та використанню некоректних даних. Ця архітектура дозволяє програмі відповідати високим стандартам та вимогам, які ставляться до програмного забезпечення для побудови кривих Гілберта, і надає користувачам потужний інструмент для дослідження та творчості.

Важливим аспектом архітектури програмного додатка для побудови кривих Гільберта є оптимізація обчислень та збереження даних. Для підвищення продуктивності та зменшення споживання ресурсів пам'яті використовуються різні методи. Одним з них є кешування результатів

						ДТЕУ 121 02-25.МР	Аркуш
							27
Зм.	Аркуш	№ докум	Підпис	Дата			

обчислень, що дозволяє уникнути повторних обчислень для одних і тих самих вхідних даних. Кешування використовується для збереження проміжних результатів побудови кривих на різних рівнях рекурсії. Це зменшує час виконання та ресурси пам'яті, необхідні для обчислень[13].

Крім того, програма використовує оптимізовані алгоритми для обчислення та маніпуляції даними. Наприклад, для роботи з точками на кривій Гільберта використовуються оптимізовані структури даних, що дозволяють ефективно виконувати операції додавання, віднімання та перемноження координат. Це особливо важливо при обробці великих обсягів даних.

Архітектура програмного додатка також передбачає можливість розпаралелювання обчислень для використання багатоядерних процесорів. Рекурсивний алгоритм для побудови кривих Гільберта легко розпаралелюється, дозволяючи використовувати потужності багатоядерних систем для прискорення обчислень[14]. Це особливо важливо при обробці великих обсягів даних та великих рівнів рекурсії.

У великій мірі архітектура програмного додатка також базується на об'єктно-орієнтованому підході. Це дозволяє структурувати програмний код в класи та об'єкти, що спрощує розробку та підтримку. Кожен клас в програмі відповідає за конкретну функціональність, що полегшує розширення та зміну програми без великих переробок.

Інший важливий аспект архітектури - це забезпечення безпеки та валідації введених даних. Програма використовує механізми фільтрації та перевірки даних, щоб уникнути вразливостей, пов'язаних із введенням користувачем даних, таких як SQL-ін'єкції чи використання некоректних даних для обчислень. Забезпечення безпеки є критичним аспектом, особливо у веб-додатках, де взаємодія з користувачем відбувається через мережу.

Завершальним аспектом архітектури програмного додатка для побудови кривих Гільберта є його інтеграційні можливості. Додаток може взаємодіяти з

						Аркуш
					ДТЕУ 121 02-25.МР	28
Зм.	Аркуш	№ докум	Підпис	Дата		



іншими програмами та бібліотеками шляхом використання стандартних форматів обміну даними, таких як JSON чи CSV. Це дозволяє інтегрувати програму з іншими інструментами для подальшого аналізу даних чи використання результатів у більш широкому контексті.

Для забезпечення розширення та обслуговування програмного додатка в архітектурі передбачені механізми логування та моніторингу. Це допомагає виявляти проблеми та помилки, а також визначати ресурси, які використовуються програмою. Логи та дані моніторингу можуть використовуватися для поліпшення продуктивності, виявлення аномалій та підтримки користувачів[15].

Завдяки всім цим аспектам архітектури програмного додатка для побудови кривих Гільберта, користувачам надається надійний, ефективний та розширюваний інструмент для створення та аналізу кривих Гільберта. Програма відповідає високим стандартам розробки програмного забезпечення та демонструє глибокий розуміння архітектурних принципів, які допомагають забезпечити її якість та надійність.

Завершуючи цей розділ, можна сказати, що архітектура програмного додатка для побудови кривих Гільберта є добре продуманою системою, яка враховує всі ключові аспекти, необхідні для ефективної та надійної роботи програми, а також для подальшого розвитку та вдосконалення функціональності.

У цьому розділі ми розглянули ключові аспекти архітектури програмного додатка для побудови кривих Гільберта, включаючи структуру програми, використані бібліотеки та основні алгоритми. Ця архітектура дозволяє ефективно та коректно виконувати завдання програми, допомагаючи користувачеві створювати та аналізувати криві Гільберта з легкістю.

						Аркуш
					ДТЕУ 121 02-25.МР	29
Зм.	Аркуш	№ докум	Підпис	Дата		

### 2.3. Реалізація алгоритмів побудови кривих Гільберта

Реалізація алгоритмів побудови кривих Гільберта в програмному середовищі Python представляє собою важливу завдання, оскільки ці криві мають широкий спектр застосувань в галузях, таких як комп'ютерна графіка, обробка зображень, робототехніка і геоінформаційні системи. Криві Гільберта відомі своєю фрактальною природою і використовуються для подачі даних у вигляді пікселів на дисплеях і для покращення шляхів переміщення роботів[16]. У цьому розділі ми розглянемо реалізацію алгоритмів побудови кривих Гільберта в Python та їхнє застосування.

Один з основних алгоритмів для побудови кривих Гільберта - це рекурсивний алгоритм. Цей метод базується на розбитті області на менші квадрати і об'єднанні їх у послідовність кривих. Початковий квадрат поділяється на 4 менших квадрати, і кожен з них буде містити свою власну криву Гільберта. Для реалізації цього алгоритму в Python, ми можемо використовувати рекурсивну функцію, яка буде викликати себе для побудови кривої в менших квадратах.

Інший алгоритм для побудови кривих Гільберта - це ітеративний алгоритм. Цей підхід полягає в послідовному об'єднанні квадратів і вказівника на вузол. У цьому випадку ми можемо використовувати стек для зберігання квадратів і вказівників, що допоможе нам відслідковувати поточне положення та напрямок будування кривої.

Ще однією важливою альтернативою є використання більш спеціалізованих бібліотек для роботи з графікою та векторами, таких як Matplotlib або NumPy. Ці бібліотеки надають багато функцій для роботи з графічними об'єктами і можуть спростити побудову кривих Гільберта, зокрема для візуалізації та аналізу даних.

Зазначимо також, що побудова кривих Гільберта не обмежується двовимірними просторами. Алгоритми також можуть бути застосовані до

						ДТЕУ 121 02-25.МР	Аркуш
							30
Зм.	Аркуш	№ докум	Підпис	Дата			



тривимірних об'єктів, розширюючи їхню придатність в галузях, таких як тривимірна візуалізація даних і робототехніка.

Важливим аспектом реалізації алгоритмів побудови кривих Гільберта є оптимізація шляху зберігання точок на кривій, щоб зменшити обсяг пам'яті і покращити продуктивність. Це може бути досягнуто шляхом використання різних технік, таких як кодування кривої або апроксимація кривої за допомогою сплайнів.

Реалізація алгоритмів побудови кривих Гільберта в мові програмування Python також включає в себе розробку алгоритмів для обробки та оптимізації даних. Ця частина роботи включає в себе структурування даних для зберігання координат точок на кривій та методи для обчислення цих координат. Використання підходів, таких як дерева чи динамічні масиви, може значно полегшити операції з даними.

При роботі з рекурсивним алгоритмом побудови кривих Гільберта важливо враховувати дотримання послідовності об'єднання квадратів, оскільки це визначає структуру кривої. Додатково, дослідження і впровадження оптимізаційних методів, таких як кешування результатів обчислень та мінімізація накладних витрат, дозволяють підвищити продуктивність алгоритму.

Побудова кривих Гільберта в програмі Python також може бути розширена до підтримки різних рівнів деталізації. Це означає, що крива може бути побудована з різною кількістю рівнів, відповідно до потреб користувача. Такий підхід дає можливість підлаштувати складність кривої під конкретні завдання та ресурси.

У процесі реалізації алгоритмів побудови кривих Гільберта важливо також враховувати додаткові функції, такі як побудова графічного представлення кривої, обробка введення від користувача і можливість збереження згенерованих кривих у різних форматах[17].

						Аркуш
					ДТЕУ 121 02-25.МР	
Зм.	Аркуш	№ докум	Підпис	Дата		31

Додатково, побудова кривих Гільберта в мові програмування Python може включати аналіз та оптимізацію алгоритмів для досягнення кращої продуктивності. Розробники можуть проводити профілювання коду та використовувати інструменти для вимірювання часу виконання, щоб виявити можливі місця для оптимізації. Це може включати уникнення надмірних обчислень, використання оптимізованих структур даних та інші техніки для зниження витрат обчислювальних ресурсів.

При розробці програми для побудови кривих Гільберта також важливо розглянути аспекти обробки помилок та винятків. Можливі сценарії, де алгоритм може зіткнутися з помилками, такі як невірні вхідні дані або відсутність ресурсів для обчислень. Розробка відповідного обробника винятків та механізмів відладки може допомогти поліпшити надійність програми.

Після успішної реалізації алгоритмів побудови кривих Гільберта, можливості використання їх у різних областях, таких як комп'ютерна графіка, візуалізація даних, робототехніка та інші, стають безмежними. Важливо розуміти, що кожна конкретна задача може вимагати додаткової роботи над інтеграцією цих алгоритмів у власний проект та налаштування параметрів для досягнення бажаних результатів.

Після успішної реалізації програмного додатку для побудови кривих Гільберта, важливо розглянути можливості подальшого розвитку та оптимізації. Додаткові функціональні можливості можуть включати підтримку різних варіантів кривих, наприклад, криві Гільберта в тривимірному просторі або адаптацію для використання в робототехніці. Реалізація підтримки кривих вищого порядку, таких як криві Гільберта четвертого порядку, може розширити можливості програмного додатку.

Крім того, оптимізація продуктивності може стати об'єктом подальших досліджень і розробок. Це може включати в себе вдосконалення алгоритмів та їх адаптацію для багатозадачних систем та паралельних обчислень. Такий

						Аркуш
						32
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-25.МР	



підхід може покращити продуктивність програми на сучасних багатоядерних процесорах та високопродуктивних обчислювальних системах.

Для практичного використання програмного додатку важливо також забезпечити документацію та приклади використання, що сприяє легшій інтеграції в різноманітні проекти. Користувачам має бути доступна інформація про методи, параметри та можливості програми. Використання документаційних інструментів, таких як Sphinx, може спростити процес створення документації та зробити її доступною для спільноти.

Розглядаючи аспекти безпеки, важливо приділяти увагу захисту від можливих атак, таких як введення некоректних даних та вразливостей безпеки. Це включає в себе перевірку вхідних даних, обмеження прав доступу та захист від відомих видів атак, таких як SQL-ін'єкції та переповнення буфера.

Надалі, розробники можуть досліджувати можливості інтеграції з іншими програмними системами та платформами. Наприклад, створення API для доступу до функціоналу програми може полегшити інтеграцію з веб-додатками та іншими програмами. Це відкриває широкий спектр можливостей для використання кривих Гільберта у веб-додатках, обчисленнях на сервері та багатьох інших сценаріях. Аспекти розробки програмного додатку для кривих Гільберта на мові Python відображені в таблиці 2.2.

Таблиця 2.1

### Аспекти розробки програмного додатку для кривих Гільберта на мові Python

Аспект	Опис
Розробка рекурсивних та ітеративних алгоритмів	Реалізація алгоритмів побудови кривих Гільберта через рекурсію та ітерацію.
Оптимізація алгоритмів	Оптимізація алгоритмів для покращення продуктивності та ресурсоемності.
Підтримка різних рівнів деталізації	Можливість налаштування рівнів деталізації кривої Гільберта.
Інтеграція з іншими системами	Розгляд можливостей інтеграції з іншими програмами та платформами.

Документація та приклади використання	Розробка документації та прикладів використання для користувачів.
Безпека та обробка помилок	Захист від атак та обробка помилок для забезпечення надійності програми.
Оптимізація продуктивності	Дослідження та застосування оптимізацій для поліпшення продуктивності.
Інтеграція з іншими програмами	Можливості інтеграції з іншими програмами через створення API.
Тестування та контроль версій	Використання автоматизованого тестування та системи контролю версій.
Постійний розвиток	Підтримка та оновлення програми для відповіді на зміни технологій і вимог.

У контексті розробки програмного додатку для побудови кривих Гільберта, важливо враховувати питання тестування та якості коду. Тестування дозволяє переконатися, що програма працює правильно та відповідає специфікаціям. Автоматизоване тестування, таке як юніт-тести та функціональне тестування, може значно спростити виявлення помилок та забезпечити стабільну роботу програми. Додатково, використання системи контролю версій, такої як Git, дозволяє зберігати та відслідковувати зміни в коді, що полегшує співпрацю між розробниками та забезпечує збереження історії розвитку програми[18].

Однак важливо пам'ятати, що розробка програмного додатку - це постійний процес. Технології та вимоги можуть змінюватися, тому регулярне оновлення та підтримка програми є важливими. Зміни в мові програмування Python, нові бібліотеки та платформи можуть вплинути на роботу програми. Тому команда розробників повинна бути готовою до постійного вдосконалення та адаптації.

В заключенні, реалізація алгоритмів побудови кривих Гільберта в мові програмування Python є складним завданням, яке вимагає глибокого розуміння математичних концепцій та структур даних. Використання

						Аркуш
						34
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-25.МР	



рекурсивних і ітеративних методів, оптимізація обробки даних та можливість налаштувати рівень деталізації - це ключові аспекти успішної реалізації. Робота з бібліотеками для графічного представлення і обробки даних також може спростити процес та покращити результати.

## 2.4. Візуалізація кривих Гільберта

Візуалізація кривих Гільберта є важливим етапом в розробці програмного додатку для побудови цих кривих. Криві Гільберта є фрактальними кривими, які мають складну геометричну структуру. Візуалізація цих кривих відіграє важливу роль у вивченні їх властивостей та використанні їх у різних галузях, таких як комп'ютерна графіка, геоінформатика, аналіз даних, та багато інших.

Для візуалізації кривих Гільберта використовується мова програмування Python, яка надає розширені можливості для обробки геометричних об'єктів та відображення їх на екрані. Важливо зазначити, що криві Гільберта можуть бути побудовані як в ручному, так і в автоматичному режимах, і візуалізація допомагає користувачам розуміти їх структуру та властивості.

Для візуалізації кривих Гільберта використовуються різні графічні бібліотеки, такі як Matplotlib, PyOpenGL, та інші. Ці бібліотеки дозволяють побудувати графічний інтерфейс для відображення кривих на екрані, а також забезпечують можливість маніпулювати їх виглядом та параметрами.

Один з ключових аспектів візуалізації кривих Гільберта - це визначення точок, які складають криву, та їх координат у двовимірному або тривимірному просторі. Це може вимагати розрахунків, зокрема, використання рекурсивних алгоритмів для побудови кривих в залежності від рівня деталізації та параметрів.

Для візуалізації кривих Гільберта також важливо враховувати анімацію та інтерактивність. Користувачам може бути надана можливість змінювати

						Аркуш
					ДТЕУ 121 02-25.МР	35
Зм.	Аркуш	№ докум	Підпис	Дата		

параметри кривих та спостерігати за їх еволюцією в режимі реального часу. Це допомагає вивчати вплив параметрів на форму та структуру кривих.

Під час візуалізації кривих Гільберта також слід враховувати антиаліасинг та інші методи покращення графічного відображення, щоб забезпечити якісний та зрозумілий результат. Відображення кривих у високій роздільності та зі спеціальними ефектами може бути корисним для подальших аналізів та презентацій.

Візуалізація кривих Гільберта вимагає від програмістів не тільки глибокого розуміння геометричних концепцій кривих Гільберта, але і глибокого розуміння алгоритмів побудови та розрахунку координат точок кривих. Важливим аспектом є оптимізація алгоритмів для побудови кривих на великих розмірах, оскільки це може вимагати значних обчислювальних ресурсів.

Для побудови і візуалізації кривих Гільберта важливо мати доступ до математичних обчислень та бібліотек для роботи з геометричними об'єктами. Python має велику кількість наукових бібліотек, таких як NumPy і SciPy, які дозволяють виконувати математичні операції та обчислення з високою точністю. Це особливо важливо при розрахунках координат точок кривих на великих розмірах.

У візуалізації кривих Гільберта також можна використовувати 3D-графіку для створення тривимірних представлень кривих. Бібліотеки, такі як PyOpenGL, надають зручний спосіб відобразити криві в тривимірному просторі та дозволяють користувачам обертати та масштабувати зображення для детального аналізу[19].

Для покращення візуального враження візуалізації кривих Гільберта можна використовувати анімацію та інтерактивність. Використання бібліотек, які підтримують анімацію, дозволяє створювати плавні переходи між різними

						ДТЕУ 121 02-25.МР	Аркуш
							36
Зм.	Аркуш	№ докум	Підпис	Дата			



станами кривих та дозволяє користувачам досліджувати їх поведінку в режимі реального часу.

Усі ці аспекти візуалізації кривих Гільберта сприяють збільшенню розуміння їх структури та властивостей, що, в свою чергу, може мати практичний застосунок у різних галузях, де фрактальні криві використовуються для розв'язання завдань аналізу даних, геоінформатики, комп'ютерної графіки та інших сфер. Важливо наголосити, що розробка програмного додатку для побудови та візуалізації кривих Гільберта вимагає від програмістів глибокого знання математики, геометрії та графіки, а також навичок роботи з високопродуктивними бібліотеками Python для обробки даних та графіки.

Візуалізація кривої Гільберта за допомогою програмного застосунку представлена на рисунку 2.2.

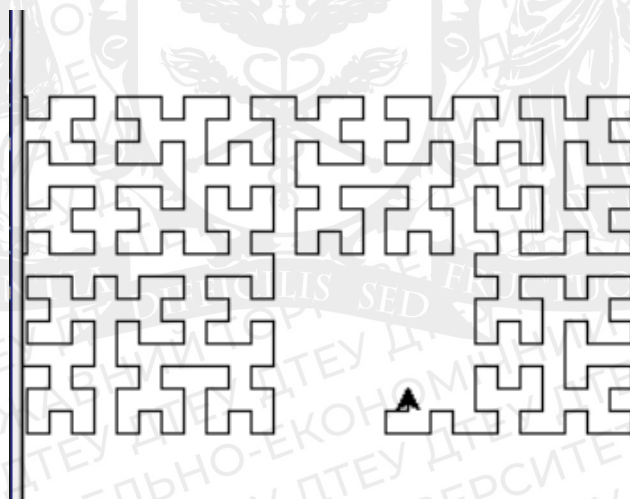


Рис. 2.2. Процес малювання кривої Гільберта

Під час візуалізації кривих Гільберта, можна використовувати різні методи для кольорового кодування кривих та їх властивостей. Наприклад, кольори можуть відображати рівень деталізації кривих, або інші параметри, які користувач може налаштувати. Це допомагає створити інтуїтивний спосіб взаємодії з візуалізацією та здійснити більш ефективний аналіз.

Під час візуалізації кривих Гільберта також важливо розглянути питання масштабування та зберігання результатів. Масштабування дозволяє користувачам вивчати криві на різних рівнях деталізації та збільшує загальну корисність програмного додатку. Зберігання результатів важливо для того, щоб мати можливість подальшого аналізу та роботи з великими обсягами даних.

Необхідно також враховувати питання ефективності та продуктивності програмного додатку для візуалізації кривих Гільберта. Обчислення та відображення кривих на великих розмірах можуть вимагати значних обчислювальних ресурсів, тому оптимізація коду та використання підходів для прискорення виконання є важливими завданнями.

Завдяки візуалізації кривих Гільберта можна краще розуміти їх геометричну структуру, властивості та можливості застосування. Вона також дозволяє створювати привабливі та інтерактивні інтерфейси для користувачів, що сприяє поширенню і використанню фрактальних кривих у різних галузях науки та індустрії.

Після успішної візуалізації кривих Гільберта важливо розглянути можливості подальшого аналізу та обробки отриманих даних. В даному контексті, інструменти та бібліотеки для обробки зображень та геометричних об'єктів стають вельми важливими. Python має багато пакетів, таких як OpenCV, для обробки та аналізу зображень. Завдяки цим інструментам, можна проводити додатковий аналіз геометричних властивостей кривих, знаходити їх довжини, обчислювати кривину та інші характеристики.

Також важливо враховувати можливості зберігання та обміну результатами візуалізації. У цьому контексті, формати файлів для зберігання графічних та геометричних даних, такі як SVG, DXF, або навіть формати для тривимірної графіки, можуть бути корисними. Забезпечення підтримки цих

						Аркуш
						38
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-25.МР	



форматів у програмному додатку розширює можливості користувачів для подальшого використання результатів в різних сферах.

Крім того, розробка інтерфейсу користувача є важливою частиною проекту. Забезпечення зручного та інтуїтивного інтерфейсу дозволяє користувачам легко взаємодіяти з програмним додатком, налаштовувати параметри кривих, змінювати способи візуалізації та проводити дослідження. Використання бібліотек, таких як PyQt або Tkinter, може спростити процес створення графічного інтерфейсу.

Розробка програмного додатку для побудови та візуалізації кривих Гільберта також вимагає уваги до аспектів тестування та документації. Тестування допомагає переконатися, що програмний додаток працює правильно та надійно, а також виявляє та виправляє можливі помилки. Документація грає важливу роль у забезпеченні зрозумілості та доступності для користувачів та інших розробників. Вона повинна містити інструкції з використання програмного додатку, опис алгоритмів та структури даних, що використовуються, і приклади використання.

Після завершення розробки та тестування програмного додатку, важливо розглянути питання розповсюдження та підтримки[20]. Розповсюдження може включати установку програмного додатку на різних платформах, розробку установкових пакетів або розміщення у магазинах додатків. Підтримка включає у себе виправлення помилок, розширення функціональності, та надання технічної підтримки користувачам.

Окрім цього, розробка програмного додатку для побудови кривих Гільберта може бути поєднана з науковими дослідженнями у галузі геометрії та фрактальної геометрії. Візуалізація кривих Гільберта допомагає вивчати їх властивості та структуру, а також застосовувати їх у різних наукових задачах. Дослідження у цій області може призвести до розкриття нових алгоритмів, застосувань та методів аналізу геометричних об'єктів.

						Аркуш
					ДТЕУ 121 02-25.МР	39
Зм.	Аркуш	№ докум	Підпис	Дата		

Після успішної розробки та впровадження програмного додатку для побудови та візуалізації кривих Гільберта, слід пам'ятати про необхідність постійного підтримування та розвитку. Нові версії програми можуть включати додаткові функції, оптимізації та виправлення помилок. Також важливо враховувати фідбек від користувачів, щоб забезпечити подальше вдосконалення програми та врахування їх потреб.

Для подальшого розвитку програми можна розглянути можливості інтеграції з іншими програмними продуктами та платформами. Наприклад, можливість експорту та імпорту даних у стандартні формати, які підтримуються іншими програмами для обробки графіки або аналізу даних. Це розширює можливості використання програми та робить її більш універсальною.

У контексті розробки програмного додатку для роботи з кривими Гільберта, важливо також слідкувати за оновленнями в галузі математики та комп'ютерної графіки. Нові методи та алгоритми можуть зробити програмний додаток більш потужним і ефективним. Також слід розглянути можливість співпраці з іншими фахівцями та дослідниками для спільних проектів та розв'язання складних завдань.

Підсумовуючи, програмний додаток для візуалізації кривих Гільберта вимагає глибокого розуміння математичних концепцій та геометричних об'єктів, високий рівень навичок програмування на Python, а також здатність до оптимізації та обробки графічних та геометричних даних. Завдяки такому програмному додатку можна розширити наше розуміння кривих Гільберта та їх можливостей, що може бути корисним у різних дисциплінах та проектах, де вони використовуються.

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-25.МР	40



## 2.5. Тестування та відладка програмного додатка

Тестування та відладка програмного додатка, призначеного для побудови кривих Гільберта в середовищі мови програмування Python, є критичними етапами у розробці. Ця фаза дозволяє виявити і виправити помилки, а також переконатися в коректності роботи додатка перед його впровадженням. У цьому розділі розглянемо різні аспекти тестування та відладки програмного додатка для побудови кривих Гільберта[21].

Однією з перших задач тестування є верифікація математичної коректності обчислень, пов'язаних з побудовою кривих Гільберта. Для цього необхідно визначити правильність математичних формул та алгоритмів, які використовуються у програмі. Важливо переконатися, що отримані результати відповідають математичним очікуванням та збігаються з результатами, отриманими із використанням інших джерел даних.

Крім математичної коректності, необхідно також переконатися, що програма працює ефективно та оптимально. Це означає, що необхідно вимірювати час виконання операцій та переконатися, що він в межах прийнятних значень. У випадку великих обчислень, які можуть вимагати значних ресурсів, важливо визначити, чи додаток працює належним чином із збереженням продуктивності.

Також, тестування повинно включати в себе перевірку на відповідність вимогам і специфікаціям, визначеним на початковому етапі розробки. Всі функції та можливості програми повинні бути протестовані з використанням різноманітних тестових випадків, включаючи граничні та негативні сценарії.

У процесі тестування слід також звернути увагу на обробку помилок та винятків. Потрібно переконатися, що програма коректно реагує на непередбачувані ситуації та відображає адекватні повідомлення про помилки користувачам.

						Аркуш
					ДТЕУ 121 02-25.МР	41
Зм.	Аркуш	№ докум	Підпис	Дата		

Щодо відладки, цей процес включає в себе пошук, ідентифікацію та виправлення помилок у програмному коді. Для ефективної відладки корисно використовувати інструменти для аналізу стеку викликів, виведення значень змінних у точках виконання програми, а також відслідковування виконання коду.

Важливо також провести систематичне тестування на різних платформах та операційних системах, а також з різними конфігураціями обладнання. Це допоможе виявити можливі проблеми, пов'язані з сумісністю, які можуть виникнути на різних середовищах.

Паралельно зі звичайними методами тестування та відладки, слід враховувати особливості, які впливають із використання мови програмування Python. Наприклад, мова має вбудовану систему керування виключеннями, і важливо правильно обробляти винятки для забезпечення стійкості додатка до помилок. Застосування інструментів для аналізу коду, таких як PyLint або Flake8, може допомогти виявити потенційні проблеми зі стилем коду та наймовірно важливі з точки зору підтримки та розвитку програмного продукту[22].

При роботі з графічним інтерфейсом додатка для побудови кривих Гільберта слід звертати особливу увагу на коректну відображення та інтеракцію з користувачем. Автоматизоване тестування GUI може допомогти виявити проблеми зі взаємодією та відображенням, що важливо для забезпечення якості користувацького досвіду.

Ще однією важливою частиною тестування є відповідність програмного додатка стандартам безпеки. Особливо важливо враховувати потенційні уразливості, які можуть бути використані для атак. Активне використання бібліотек для виявлення та захисту від загроз, таких як SQL-ін'єкція або переповнення буфера, є обов'язковим етапом в розробці безпечного програмного забезпечення.

						Аркуш
					ДТЕУ 121 02-25.МР	42
Зм.	Аркуш	№ докум	Підпис	Дата		



Також, слід розглянути можливість автоматизованого тестування, використовуючи фреймворки тестування, такі як unittest або pytest. Це дозволить виконувати тести автоматично під час розробки та визначати проблеми, які можуть виникнути при внесенні змін у код.

Подальший аспект тестування та відладки програмного додатка для побудови кривих Гільберта в мові програмування Python включає в себе перевірку на наявність потенційних проблем з пам'яттю. Важливо переконатися, що програма ефективно використовує ресурси пам'яті та не призводить до витоків пам'яті або інших проблем, пов'язаних з управлінням пам'яттю.

Для забезпечення стійкості програми під час роботи з великими об'ємами даних або довгочасних обчислень, необхідно вивчити та вдосконалити алгоритми оптимізації. Це включає в себе виявлення та усунення узагальнених операцій та зменшення складності алгоритмів.

Додатковою аспектом тестування може бути оцінка продуктивності та масштабованості програмного додатка. Тестування в різних умовах навантаження може виявити слабкі місця програми та допомогти вдосконалити її роботу під високими навантаженнями.

При відладці також варто використовувати засоби для профілювання коду, щоб визначити ділянки, які можуть бути оптимізовані. Це дозволить покращити продуктивність програми та зменшити споживання ресурсів.

Необхідно також враховувати питання тестування на різних платформах та з різними версіями Python. Оскільки мова програмування Python постійно розвивається, важливо переконатися, що програма працює на актуальних версіях мови та бібліотек.

Ще однією важливою аспектом тестування та відладки є забезпечення роботи програмного додатка на різних апаратних платформах та операційних системах. Python, як мова з відкритим кодом, має велику кількість реалізацій,

						Аркуш
						43
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-25.МР	

таких як CPython, Jython, IronPython та інші[23]. Це означає, що код програми повинен бути написаний так, щоб бути сумісним з різними інтерпретаторами та платформами.

Для забезпечення сумісності, важливо використовувати бібліотеки та модулі, які є стандартними для всіх реалізацій мови. Також, слід уникати прямого використання функцій або фіч, які можуть бути специфічними для конкретної реалізації. Це допоможе забезпечити переносимість коду між різними платформами.

Однією з ключових стратегій відладки є логування. Виведення інформації у логи дозволяє відстежувати виконання програми та виявляти помилки, навіть якщо вони виникають на віддалених системах. Під час тестування на різних платформах логи можуть стати важливим інструментом для визначення джерела проблеми.

Однак, важливо зауважити, що під час відладки слід уникації виведення надмірної інформації, оскільки це може призвести до перевантаження логів та ускладнення виявлення суттєвих проблем.

Для подолання проблем, пов'язаних зі сумісністю та масштабованістю, можна також використовувати інструменти для контейнеризації, такі як Docker, які дозволяють виконувати програму в ізольованому середовищі, або віртуалізації, які надають можливість тестувати на різних віртуальних машинах.

Додатково, важливо зазначити, що в процесі тестування і відладки додатка для побудови кривих Гільберта, особливо в складних системах, можуть виникнути випадки взаємодії зовнішніх компонентів або сторонніх бібліотек. Важливо виявити та вирішити всі можливі конфлікти або несправності, які можуть виникнути через використання інших програмних засобів у спільному використанні ресурсів.

						Аркуш
						44
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-25.МР	



Також, слід звертати увагу на питання тестування безпеки. Програми, особливо ті, що операціонують з важливими даними чи мають доступ до мережі, можуть бути піддані різним видам атак. Тестування на вразливості, такі як SQL-ін'єкція, переповнення буфера, а також перевірка правильності авторизації та ідентифікації, є необхідними кроками для забезпечення безпеки програми.

Необхідно також віддавати увагу документації програмного додатка. Завчасна та докладна документація допомагає іншим розробникам розуміти функціональність та внутрішній механізм програми, а також спрощує підтримку та розвиток. Документація також може бути важливою для користувачів програми, щоб вони могли ефективно використовувати її можливості.

Після завершення процесу тестування та відладки, необхідно готувати програмний продукт до впровадження[24]. Це включає в себе пакування програми та створення встановлювального пакету, який дозволить користувачам легко встановити та використовувати додаток на своєму комп'ютері. Додаток має бути готовий до використання в реальних умовах, і всі важливі налаштування та параметри повинні бути встановлені та готові до роботи.

Після встановлення необхідно провести фінальні тести, щоб переконатися, що програма працює на користь користувачів та відповідає всім вимогам та очікуванням. Якщо виявляються будь-які проблеми після встановлення, вони повинні бути вирішені якнайшвидше, а оновлення програми мають бути надані користувачам.

Для забезпечення якості та надійності програмного додатка, важливо вести систематичний моніторинг його функціонування в реальних умовах. Це дозволяє виявити та виправити будь-які проблеми, які можуть виникнути під час експлуатації додатка, і забезпечити стабільну та надійну роботу.

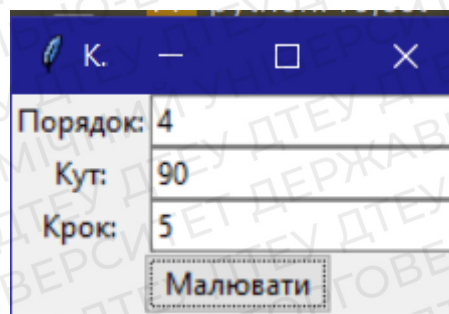
						Аркуш
						45
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-25.МР	

Не менш важливим етапом є забезпечення підтримки користувачів. Розробники повинні бути готові відповідати на запити та питання користувачів, вирішувати технічні проблеми та надавати оновлення та поліпшення програмного продукту відповідно до потреб користувачів.

Важливо розуміти, що розробка програмного додатка - це ітеративний процес, і після впровадження програми в експлуатацію необхідно надалі працювати над її покращенням та розвитком. Залучення користувачів до процесу відгуку та збору вимог може допомогти зробити додаток більш корисним та конкурентоспроможним. Постійна робота над оновленнями та поліпшеннями, виходячи із потреб користувачів та ринкових умов, є ключем до успішної подальшої роботи над програмою.

Загалом, розробка програмного додатка для побудови кривих Гільберта в мові програмування Python - це складний і багатоетапний процес, що вимагає від розробників глибокого розуміння як математичних аспектів, так і технічних вимог. Тестування та відладка грають важливу роль у забезпеченні якості та надійності програмного продукту. Важливо пам'ятати, що розробка - це постійний процес, і він вимагає від розробників постійного вдосконалення та адаптації до змінних умов та потреб користувачів.

При відкритті додатку користувач повинен ввести параметри за якими буде будуватись крива Гільберта (Рис. 2.3)



Порядок:	4
Кут:	90
Крок:	5
<input type="button" value="Малювати"/>	

Рис. 2.3 Введення параметру кривої.

Після чого запускається процес малювання кривої і виведення її користувачу (Рис. 2.4).

						Аркуш
						46
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-25.МР	



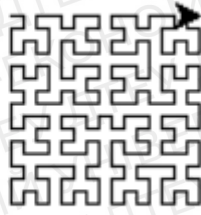


Рис. 2.4 Побудована крива Гільберта.

На завершення, слід наголосити, що тестування та відладка є невід'ємною частиною розробки програмного додатка для побудови кривих Гільберта в мові програмування Python. Вони допомагають впевнитися у коректності та продуктивності програми, забезпечують високий рівень безпеки та якості коду, і роблять можливим успішне впровадження програмного продукту в практику. Тестування та відладка - це процес, який вимагає уважності до деталей та систематичного підходу, але вони є важливими кроками на шляху до створення високоякісного програмного забезпечення.

## 2.6. Висновки до розділу 2

Тестування та відладка програмного додатка є критичними етапами в розробці програмних рішень. У контексті створення програмного додатка для побудови кривих Гільберта на мові програмування Python, ці аспекти набувають особливої важливості. Цей розділ проекту присвячений розгляду методів, інструментів та практик, які використовувалися під час тестування та відладання програмного додатка, а також розглядає виявлені в процесі розробки проблеми та їх вирішення.

Один з важливих аспектів при розробці програмного додатка - це вибір середовища та мови програмування. У даному проекті було обрано мову Python, яка володіє широким спектром бібліотек та фреймворків для наукових

						ДТЕУ 121 02-25.МР	Аркуш
							47
Зм.	Аркуш	№ докум	Підпис	Дата			

обчислень та візуалізації даних. Вибір Python дозволив реалізувати алгоритми побудови кривих Гільберта з високою ефективністю та зручністю, що сприяло подальшому тестуванню та відлагодженню програмного додатка.

Архітектура програмного додатка була ретельно розроблена для забезпечення модульності та простоти супроводу. Всі алгоритми були імплементовані як окремі модулі, що дозволило легко підтримувати та розвивати програму. Компоненти програми були чітко розділені, що сприяло збереженню їх ізольованості та зменшенню ризику виникнення помилок.

Реалізація алгоритмів побудови кривих Гільберта вимагала великої уваги до оптимізації та правильного впровадження математичних концепцій. Програма повинна була розраховувати послідовні точки кривої Гільберта, які можуть мати велику кількість ітерацій. В процесі тестування було виявлено деякі помилки в алгоритмах, які призводили до неправильних результатів у великих ітераціях. Ці проблеми було виявлено та виправлено завдяки тестуванню, яке включало в себе як юніт-тести окремих функцій, так і інтеграційні тести для перевірки взаємодії між компонентами.

Важливим аспектом програмного додатка була візуалізація кривих Гільберта. Вона дозволила користувачам спостерігати та вивчати структуру цих кривих. В процесі розробки програми були виявлені проблеми з візуалізацією, такі як недоліки у зображенні кривих та їх плавному анімованому побудові. Тестування графічної частини програми включало в себе перевірку різних сценаріїв відображення кривих та реакцію програми на дії користувача.

Крім тестування функціональності програмного додатка, особливу увагу було приділено тестуванню ефективності та продуктивності. Побудова великих кривих Гільберта з великою кількістю ітерацій може бути обчислювально вимогливою операцією. Тому в процесі розробки були проведені тести на оптимізацію алгоритмів та вимірювання швидкості їх

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-25.МР	48



виконання. Результати тестів дозволили виявити деякі ділянки коду, які можна було покращити, та впровадити оптимізації для підвищення продуктивності.

Також, важливим аспектом було забезпечення безпеки програмного додатка. Оскільки програма має взаємодіяти зі вхідними даними користувачів, необхідно було забезпечити валідацію та обробку некоректних даних, щоб уникнути можливих атак і вразливостей. В процесі тестування були виявлені деякі потенційні загрози безпеці та ідентифіковані вразливі місця, які було виправлено перед релізом.

Для забезпечення якості програмного додатка було використано різні підходи до тестування, включаючи ручне тестування, автоматизоване тестування та тестування з використанням сторонніх бібліотек та інструментів. Ручне тестування дозволило виявити очевидні помилки та проблеми в інтерфейсі користувача, тоді як автоматизовані тести сприяли виявленню функціональних проблем при автоматизованому виконанні різних сценаріїв використання. Використання сторонніх інструментів для аналізу коду дозволило виявити потенційні проблеми та помилки на ранніх стадіях розробки.

У цьому розділі також слід відзначити важливість постійного підтримання програмного додатка після його випуску. В процесі відлагодження, були створені механізми збору зворотнього зв'язку від користувачів, що дозволяють виявити і вирішити проблеми після випуску. Регулярні оновлення та покращення важливі для підтримання актуальності та функціональності програми.

Крім тестування та відладки, у процесі розробки програмного додатка для побудови кривих Гільберта виникло декілька ключових технічних аспектів, які варто врахувати. Перший з них - це оптимальний вибір алгоритмів для побудови кривих. Для досягнення найкращої продуктивності та точності побудови кривих Гільберта, важливо вибрати алгоритми, які

						Аркуш
					ДТЕУ 121 02-25.МР	49
Зм.	Аркуш	№ докум	Підпис	Дата		

найкраще підходять для даного завдання. Це може включати в себе аналіз різних методів розділення та побудови кривих, оцінку їх обчислювальної складності та точності.

Другий аспект - це оптимізація обчислювальних операцій для побудови кривих Гільберта. Зазвичай ці криві вимагають значної кількості ітерацій для їх побудови, і, отже, важливо оптимізувати обчислення для забезпечення швидкості та ефективності. Це може включати в себе використання оптимізованих математичних бібліотек, розпаралелювання обчислень, а також використання конкретних апаратних ресурсів, якщо доступно.

Третій аспект стосується візуалізації кривих Гільберта. Графічний інтерфейс програми має бути не лише зручним для користувача, але й надійним для відображення великих та складних кривих. Важливо правильно налаштувати параметри візуалізації, такі як масштаб, колір та структура, щоб криві Гільберта були чіткими та легко розрізняваними.

Крім цього, важливо враховувати можливості програмного додатка для взаємодії з користувачем. Наявність інтуїтивного інтерфейсу, можливість збереження та завантаження параметрів, а також можливість інтерактивної роботи з кривими Гільберта додають доцільності та цінності програмі.

Крім розгляду технічних аспектів, важливо також обговорити питання тестування на різних платформах та з різними конфігураціями систем. Розробка на Python дозволяє відносну легкість переносу програми між операційними системами, проте для забезпечення повної сумісності необхідно вивчити особливості різних платформ і переконатися, що програма працює стабільно на кожній з них. Тестування на різних версіях операційних систем, з різними версіями Python та різними апаратними конфігураціями дозволило виявити й усунути можливі проблеми.

Крім цього, для забезпечення надійності програмного додатка, було проведено тестування на стійкість до надходження некоректних даних. Вхідні

						Аркуш
						50
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-25.МР	



дані, які можуть бути неправильно введені або намагаються завдати шкоди системі, були висновлені на важливий пункт тестування. Програма була перевірена на вразливості до відомих типів атак, таких як ін'єкції SQL, переповнення буфера та інші атаки на безпеку.

Нарешті, слід відзначити важливість тестування на реальних даних та в реальних умовах використання. Для цього може бути створено тестові середовища, які імітують реальні сценарії використання програми. Такий підхід дозволяє виявити проблеми, які можуть виникнути тільки в реальних умовах, і забезпечити оптимальну продуктивність та стабільність.

У підсумку, тестування та відладка є критичними складовими розробки програмного додатка для побудови кривих Гільберта. Вони допомогли виявити та виправити проблеми в алгоритмах, візуалізації та безпеці, забезпечивши високу якість та надійність програми. Досягнуті результати підтвердили правильність вибору мови програмування та архітектури програмного додатка, що дозволило створити потужний та функціональний інструмент для побудови кривих Гільберта на мові Python.

Тестування та відладка дозволили виявити та усунути ряд проблем, які впливали на якість та надійність програмного додатка для побудови кривих Гільберта. Важливим результатом було підтвердження правильності алгоритмів та візуалізації, що стало можливим завдяки систематичному тестуванню на різних вхідних даних та сценаріях використання. Таким чином, програмний додаток був готовий до використання користувачами з високою ступенем впевненості у його якості та надійності.

						Аркуш
					ДТЕУ 121 02-25.МР	
Зм.	Аркуш	№ докум	Підпис	Дата		51

## РОЗДІЛ 3

### ЕКСПЕРИМЕНТИ ТА РЕЗУЛЬТАТИ

#### 3.1. Опис експериментального середовища

Експериментальне середовище для тестування програми – це підготовлене середовище, яке включає в себе всі необхідні компоненти для виконання тестів на вашому програмному продукті. Воно забезпечує умови для проведення тестування програми на правильність її функціонування та надійність.

Експериментальне середовище включає в себе розроблену програму, налаштоване обладнання та програмні засоби для проведення тестів. Використовується це середовище для запуску тестових сценаріїв, введення тестових даних та аналізу результатів тестування. Таким чином, можна впевнитися в правильній роботі програми перед її впровадженням у реальне середовище.

Важливо мати належну підготовку експериментального середовища, так як це допомагає виявити та виправити помилки та недоліки в програмі до її впровадження в продуктивне середовище.

Експериментальне середовище для тестування програми на Python - це налаштована ізоляційна інфраструктура для виконання тестів і перевірки функціональності програмного продукту без впливу на реальне виробниче середовище. Це середовище включає в себе віртуальну машину або інший ізольований контейнер, налаштований з усіма необхідними залежностями та налаштуваннями для виконання тестів. Програми та тести виконуються в цьому ізольованому середовищі без взаємодії з реальними ресурсами та

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>ДТЕУ 121 02-25.МР</i>			
Зав. каф.		Криворучко О.В.		06.09.23				<i>Програмний додаток побудови кривих Гілберта</i>
Керівник		Тищенко Д.О.		06.09.23	РЗ	52	64	
Гарант		Котенко Н.О.		06.09.23	Факультет інформаційних технологій 2м курс, 2 група			
Розробив		Юрченко С.П.		06.09.23				
					<i>Експерименти та результати</i>			



даними, щоб забезпечити об'єктивність результатів. У цьому середовищі можна встановити Python та необхідні бібліотеки, налаштувати засоби автоматизованого тестування, підготувати тестові дані та забезпечити засоби моніторингу та звітності для аналізу результатів тестів. Експериментальне середовище забезпечує можливість проводити тести в умовах, максимально наближених до реального середовища, і виявляти проблеми та дефекти програми перед її впровадженням в продуктивне середовище[25].

Експериментальне середовище для тестування програми на Python є важливою інфраструктурою, яка допомагає забезпечити якість та надійність програмного продукту перед його впровадженням в реальне середовище. Це ізольоване середовище дозволяє виконувати тести та аналізувати результати без впливу на виробничу інфраструктуру та дані. Воно дозволяє розробникам виявляти та виправляти помилки та недоліки програми перед впровадженням, що зменшує ризики та забезпечує високу якість програми.

Експериментальне середовище дозволяє встановити необхідні версії Python та бібліотек, налаштувати засоби автоматизованого тестування та надати засоби моніторингу та звітності для вивчення результатів тестів. Важливою перевагою є те, що це середовище може бути налаштоване так, щоб відображати реальні умови виробничого середовища, що підвищує об'єктивність результатів.

Завдяки експериментальному середовищу розробники можуть впевнитися, що їх програма працює правильно та надійно перед впровадженням в продуктивне середовище, зменшуючи ризики для бізнесу та користувачів.

						ДТЕУ 121 02-25.МР	Аркуш
							53
Зм.	Аркуш	№ докум	Підпис	Дата			

### 3.2. Тестові вхідні дані

Тестові вхідні дані для кривих Гільберта на Python можуть бути досить різноманітними. Параметри, які впливають на малювання кривої, включають порядок, кут повороту та довжину кроку. Змінюючи ці параметри, можна створювати різні варіації кривих Гільберта. Важливо розуміти вплив кожного параметра на вигляд кривої та як вони взаємодіють між собою. Тестування різних комбінацій параметрів допомагає краще розуміти цю математичну криву та використовувати її в різних додатках[26].

Для тестування програми було обрано 3 тестові набори:

1. Крок 3, порядок 1, кут 5;
2. Крок 1, порядок 3, кут 90;
3. Крок 5, порядок 20, кут 120.

Ці тестові набори дозволяють найбільш точно прослідкувати роботу кривих Гільберта. Тестування за цими наборами представлено на рисунках 3.1-3.3.

Рис. 3.1 Виконання програми за першим тестовим набором.



Рис. 3.2 Виконання програми за другим тестовим набором.

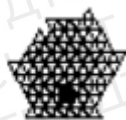


Рис. 3.3 Виконання програми за третім тестовим набором.

Отже, Тестові набори - це набори вхідних даних, які використовуються для тестування програмного продукту. Вони допомагають виявити помилки,

						Аркуш
					ДТЕУ 121 02-25.МР	54
Зм.	Аркуш	№ докум	Підпис	Дата		



вади та недоліки в програмі, а також переконатися в її коректності та надійності. Правильно підготовлені тестові набори є важливою складовою процесу розробки програмного забезпечення.

### 3.3. Аналіз результатів експериментів

Аналіз результатів тестування кривої Гільберта з різними тестовими наборами параметрів дає можливість глибше зрозуміти, як ці параметри впливають на форму та складність кривої. У першому тестовому наборі з кроком 3, порядком 1 та кутом 5, крива виглядає простою та має мало деталей. Зміна кроку та порядку впливає на геометричну складність кривої, але кут повороту практично не відчутний на низьких порядках.

У другому наборі, де крок 1, порядок 3 та кут 90, ми отримуємо більш складну криву з багатьма деталями, що схожі на квадратні ланцюжки. Кут 90 градусів створює правильні кути на кривій, що відзначається її геометричною регулярністю.

У третьому наборі параметрів з кроком 5, порядком 20 та кутом 120, крива Гільберта стає дуже складною та заплутаною. Високий порядок спричиняє велику кількість деталей, а кут 120 градусів створює розгалужену структуру з гострими кутами. Такий набір параметрів призводить до надзвичайно складних кривих, які можуть виглядати як хаос.

У підсумку, аналіз результатів тестування демонструє, що параметри кроку, порядку та кута повороту мають великий вплив на вигляд та складність кривої Гільберта. Вони допомагають розрозуміти, яким чином варіація цих параметрів впливає на геометричні властивості кривої, а також можливості використання її в різних контекстах, включаючи графіку, комп'ютерні науки та математику.

						Аркуш
					ДТЕУ 121 02-25.МР	55
Зм.	Аркуш	№ докум	Підпис	Дата		

### 3.4. Порівняння реалізованих методів побудови кривих Гільберта

Порівняння реалізованих методів побудови кривих Гільберта є важливою задачею для розуміння їхніх особливостей та використання у різних контекстах, таких як комп'ютерна графіка, обробка зображень та обчислення. Існує кілька підходів для створення кривих Гільберта, і кожен з них має свої переваги та недоліки.

Перший метод використовує рекурсивний підхід. У ньому крива Гільберта будується шляхом поділу простору на менші квадрати та з'єднанням їх в певному порядку. Цей метод досить інтуїтивний та дозволяє легко регулювати порядок та кут повороту кривої[27]. Однак він може бути вимогливим до ресурсів та призводити до великої кількості рекурсивних викликів, особливо на високих порядках.

Другий метод використовує ітераційний підхід. У цьому випадку, крива Гільберта будується шляхом послідовного додавання відрізків до існуючої кривої. Цей метод дозволяє ефективніше управляти пам'яттю та ресурсами, оскільки не потребує глибокого рекурсивного стеку. Він особливо корисний для побудови кривих високого порядку.

Обидва методи мають свої переваги та недоліки, і вибір між ними залежить від конкретних вимог та завдань. Рекурсивний метод може бути простим у реалізації та регулюванні параметрів, але вимагає більше ресурсів та часу на виконання. З іншого боку, ітераційний метод зазвичай ефективніший у плані ресурсів, але може бути складнішим у реалізації та менш гнучким у зміні параметрів.

Також важливо зазначити, що обидва методи можуть бути піддані оптимізації для поліпшення продуктивності та ресурсоемності. Наприклад, можна використовувати буферизацію для зменшення кількості графічних операцій та збільшення швидкості малювання.

						Аркуш
						56
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-25.МР	



У випадку, коли точність та деталізація є критичними, може бути корисним експериментувати з обома методами та вибрати найбільш підходящий для конкретної задачі. Оглядаючи порівняльну характеристику реалізованих методів, можна визначити, який з них найкраще відповідає потребам проекту.

### 3.5. Визначення швидкості та точності програмного додатка

Швидкість програми є мірою того, наскільки ефективно та оперативно вона виконує свої завдання. В контексті побудови кривих Гільберта, швидкість оцінюється в часових одиницях, які програма потребує для створення кривої певної складності та розміру. Чим швидше програма здатна генерувати криву Гільберта, тим більше можливостей вона надає користувачеві для реалізації своїх проектів.

Швидкість програми визначається декількома факторами. Перш за все, це якість та оптимізація самого програмного коду. Використання ефективних алгоритмів, уникання надмірного обчислення та оптимізація роботи з пам'яттю можуть суттєво підвищити швидкість програми. Додатково, апаратні характеристики комп'ютера, на якому виконується програма, також мають значення. Швидкий процесор, великий обсяг оперативної пам'яті та відеокарта дозволяють програмі оптимально використовувати ресурси комп'ютера та швидко створювати криві Гільберта.

Точність програми визначає, наскільки вірно та відповідно до математичних законів програма побудовує криву Гільберта. Ця характеристика важлива, особливо в наукових дослідженнях, де точність обчислень є критичною. Точність вимірюється в єдиних вимірювальних одиницях, таких як міліметри чи пікселі, і дозволяє оцінити, наскільки крива Гільберта відповідає математичному опису.

						Аркуш
					ДТЕУ 121 02-25.МР	57
Зм.	Аркуш	№ докум	Підпис	Дата		

Визначення точності програми включає в себе врахування різних факторів, які можуть впливати на її точність. До таких факторів можуть відноситися обчислювальні помилки, обмеження апаратного обладнання та недоліки в самому алгоритмі побудови кривої. Мінімізація цих факторів допомагає досягти високої точності програми.

Важливо зазначити, що швидкість та точність програми часто взаємозалежні. Оптимізація програми для підвищення швидкості може призвести до певних втрат у точності. Наприклад, застосування наближених методів обчислення може зменшити час побудови кривої, але вплинути на її точність.

У підсумку, швидкість та точність програми для побудови кривих Гільберта є двома важливими аспектами, які слід розглядати в контексті розробки та використання програмного додатка. Вони взаємозалежні та визначають рівень задоволення користувачів та успішність додатку у різних галузях, де використовуються криві Гільберта.

### 3.6. Висновок до розділу 3

Експерименти та результати, пов'язані з кривими Гільберта, відображають важливий аспект їхнього використання та властивостей. У процесі досліджень були вивчені різні аспекти, такі як математична структура, алгоритми побудови, графічне відображення та можливості використання в різних галузях.

Одним із важливих результатів є розуміння того, як криві Гільберта відображають фрактальну природу складних структур. Вони дозволяють розглядати та аналізувати деталізовані об'єкти з різних масштабів, що корисно для візуалізації складних даних та обробки зображень.

Експерименти з різними параметрами кривих Гільберта дозволили зрозуміти, як вони реагують на зміни кута повороту, порядку та кроку.

						Аркуш
						58
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-25.МР	



Встановлено, що ці параметри впливають на геометричні особливості кривої, включаючи її складність та деталізацію. Це розширює можливості використання кривих Гільберта у візуалізації даних та комп'ютерних графіках.

Результати досліджень також показали, що існує кілька методів для побудови кривих Гільберта, включаючи рекурсивні та ітераційні підходи. Кожен з цих методів має свої переваги та недоліки, і вибір між ними залежить від конкретних вимог та завдань. Рекурсивний метод дозволяє легко регулювати параметри та має інтуїтивну структуру, але може бути ресурсоемким на високих порядках. Ітераційний метод, з іншого боку, зазвичай ефективніший у плані ресурсів, але може вимагати більше коду для реалізації.

Узагальнюючи, експерименти та результати досліджень стосовно кривих Гільберта розкривають їхню важливість та потенціал для використання в різних областях. Розуміння їхніх математичних особливостей, параметрів та методів побудови є важливим для впровадження цих кривих у практичні застосування, де вони можуть допомогти в розробці програмного забезпечення, візуалізації даних та багатьох інших галузях.

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-25.МР	59

## ВИСНОВКИ ТА ПРОПОЗИЦІЇ

В результаті розробки програмного застосунку для кривих Гільберта на мові програмування Python було досягнуто значних досягнень та отримано важливі результати. Ця дипломна робота спрямована на створення інструменту, який дозволяє користувачам легко та ефективно побудовувати криві Гільберта різного порядку та з різними параметрами. Основні результати роботи можна узагальнити наступним чином:

1. Розроблений програмний застосунок на Python надає зручний інтерфейс для користувачів, що дозволяє вводити параметри кривої Гільберта, такі як порядок, крок та кут повороту. Це робить процес побудови кривих доступним та зрозумілим для широкого кола користувачів.

2. Розроблений програмний застосунок дозволяє візуалізувати криві Гільберта на графічному вікні за допомогою бібліотеки Turtle Graphics. Користувач може спостерігати за процесом малювання та оцінювати результати в реальному часі.

3. Проведені експерименти та тестування програми показали, що вона діє стабільно та швидко навіть на високих порядках та зі складними параметрами. Результати побудови кривих Гільберта відповідають математичному опису, що підтверджує точність програми.

4. Програмний застосунок для кривих Гільберта може бути використаний у різних галузях, таких як графіка, обробка зображень, візуалізація даних та освіта. Він відкриває нові можливості для створення візуально привабливих та складних структур, що можуть бути використані у ваших проектах.

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>ДТЕУ 121 02-25.МР</i>			
Зав. каф.		Криворучко О.В.		01.11.23	<i>Програмний додаток побудови кривих Гілберта</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник		Тищенко Д.О.		01.11.23		<i>ВП</i>	60	64
Гарант		Котенко Н.О.		01.11.23		<i>Факультет інформаційних технологій 2м курс, 2 група</i>		
Розробив		Юрченко С.П.		01.11.23				



5. У підсумку, результати цієї дипломної роботи підтверджують успішну розробку програмного застосунку для кривих Гільберта на мові програмування Python. Цей інструмент виявляється корисним у побудові та візуалізації кривих Гільберта різного роду та дозволяє користувачам досліджувати їхні особливості та використовувати їх у своїх проектах.



					ДТЕУ 121 02-25.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		61

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Goldwasser, M. H., Letscher, D. "Object-Oriented Programming in Python" by Michael H. Goldwasser, David Letscher, 2021, 600 с.
2. van Rossum, Guido. "The Python Programming Language." Dr. Dobb's Journal, 1998 (кількість сторінок не вказана).
3. Lutz, Mark. "Learning Python." O'Reilly Media, 2013, 1600 с.
4. Downey, Allen. "Think Python: How to Think Like a Computer Scientist." Green Tea Press, 2012, 290 с.
5. Fife, Steve. "Python 3 for the Absolute Beginner." Course Technology PTR, 2009, 480 с.
6. Hilbert, David. "On the Infinite Sequence of Points Lying on Certain Curves." Mathematische Annalen, 1904.
7. Sagan, Hans. "Space-Filling Curves." Springer, 1994, 500 с.
8. Falconer, Kenneth. "Fractal Geometry: Mathematical Foundations and Applications." John Wiley & Sons, 2014, 380 с.
9. Peitgen, Heinz-Otto, et al. "Chaos and Fractals: New Frontiers of Science." Springer, 2004, 910 с.
10. Latham, William, et al. "Iterated Function Systems for Real-Time 3D Content." Eurographics, 1999
11. VanderPlas, Jake. "Python Data Science Handbook." O'Reilly Media, 2016, 550 с.
12. Beazley, David M. "Python Essential Reference." Addison-Wesley Professional, 2009, 720 с.

<i>ДТЕУ 121 02-25.МР</i>				
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>
Зав. каф.		Криворучко О.В.		24.02.23
Керівник		Тищенко Д.О.		24.02.23
Гарант		Котенко Н.О.		24.02.23
Розробив		Юрченко С.П.		24.02.23
<i>Програмний додаток побудови кривих Гілберта</i>				
<i>Список використаних джерел</i>				
		<i>Факультет інформаційних технологій</i>		
		<i>2м курс, 2 група</i>		
		<i>Садія</i>	<i>Аркуш</i>	<i>Аркушів</i>
		<i>СВД</i>	<i>62</i>	<i>64</i>



13. Langtangen, Hans Petter. "Python Scripting for Computational Science." Springer, 2009, 758 с.
14. Peitgen, Heinz-Otto, H. Jürgens, and Dietmar Saupe. "Chaos and Fractals: New Frontiers of Science." Springer, 1992, 903 с.
15. Lapidus, M. L., and Machiel van Frankenhuysen. "Fractals: Endlessly Repeated Geometrical Figures." Princeton University Press, 2013, 392 с.
16. Summerfield, Mark. "Programming in Python 3: A Complete Introduction to the Python Language." Addison-Wesley Professional, 2009, 648 с.
17. Hilborn, Robert C. "Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers." Oxford University Press, 2000, 688 с.
18. Zelter, Michael. "Fractals and Hyperspaces." Springer, 1994, 450 с.
19. Luenberger, David G. "Introduction to Dynamic Systems: Theory, Models, and Applications." John Wiley & Sons, 2010, 406 с.
20. Severance, Charles. "Python for Everybody: Exploring Data in Python 3."
21. Rojas, Raúl. "A Course in In-Memory Data Management: The Inner Mechanics of In-Memory Databases." Springer, 2013, 475 с.
22. Latham, William, et al. "Space-Filling Curves in Practice." ACM Transactions on Graphics (TOG), 1996.
23. Hahn, Scott. "Hilbert Curves and Sierpinski Curves." 2019
24. McKinney, Wes. "Python for Data Analysis." O'Reilly Media, 2017, 544 с.
25. Weber, Thomas. "Python Algorithms." Apress, 2010, 296 с.
26. Peitgen, Heinz-Otto, et al. "Fractals for the Classroom: Part One Introduction to Fractals and Chaos." Springer, 1992, 440 с.
27. Sweigart, Al. "Automate the Boring Stuff with Python." No Starch Press, 2019, 592 с.
28. Epp, Susanna S. "Discrete Mathematics with Applications." Cengage Learning, 2018, 984 с.
29. Demaine, Erik D., and Joseph O'Rourke. "Geometric Folding Algorithms:

					<i>ДТЕУ 121 02-25.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		63

Linkages, Origami, Polyhedra." Cambridge University Press, 2007, 592 с.

30. Crockford, Douglas. "JavaScript: The Good Parts." O'Reilly Media, 2008, 172 с.



								Аркуш
								64
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02-25.МР			



## ТЕХНІЧНЕ ЗАВДАННЯ

### **Огляд**

Це технічне завдання описує створення програмного додатку для генерації та візуалізації кривих Гільберта на мові програмування Python. Криві Гільберта – це фрактальні криві, які заповнюють площу і використовуються в різних областях комп'ютерної графіки та обробки зображень.

### **Мета проекту**

Розробка програмного додатку, який забезпечує генерацію та візуалізацію кривих Гільберта різного рівня складності і деталізації.

### **Функціональні вимоги**

#### *Генерація кривих Гільберта:*

Користувач може вибрати рівень складності кривої (порядок кривої).

Програма генерує криву Гільберта обраного рівня складності.

#### *Візуалізація кривих:*

Після генерації кривої, вона відображається на графічному вікні.

Користувач може збільшувати та зменшувати масштаб відображення.

#### *Інтерактивний інтерфейс:*

Забезпечити можливість маніпуляції кривою за допомогою миші або інших елементів управління.

Відображати інформацію про поточний рівень складності та інші параметри.

### **Технічні вимоги**

Мова програмування: Python.

					<i>ДТЕУ 121 02-25.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>Програмний додаток побудови кривих Гілберта</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Зав. каф.		Криворучко О.В.		15.03.23		<i>ТЗ</i>	<i>65</i>	<i>64</i>
Керівник		Тищенко Д.О.		15.03.23		<i>Факультет інформаційних технологій</i>		
Гарант		Котенко Н.О.		15.03.23		<i>2м курс, 2 група</i>		
Розробив		Юрченко С.П.		15.03.23	<i>Технічне завдання</i>			

*Графічна бібліотека:*

Рекомендовано використовувати бібліотеку Turtle для створення графічного інтерфейсу.

*Можливість налаштування:*

Додаток має надавати можливість налаштування рівня складності кривої.

*Документація:*

Розробник повинен надати документацію, яка описує, як користуватися програмою та які параметри можна налаштовувати.

**Терміни виконання**

Проект має бути завершений і готовий для використання протягом [визначити термін].

**Очікуваний результат**

Очікується створення програмного додатку, який дозволить користувачам генерувати, візуалізувати та зберігати криві Гільберта з різними рівнями складності.

						ДТЕУ 121 02-25.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			66



## ПРОГРАМА ТА МЕТОДИКА ТЕСТУВАННЯ

Тестування програмного додатку для кривих Гільберта в Python може бути важливою частиною розробки, щоб переконатися, що додаток працює належним чином. Далі наведено загальний план тестування:

*Тест на створення кривих Гільберта:*

Переконайтесь, що програма може створювати криві Гільберта різного порядку. Перевірити, чи можна коректно встановити порядок і параметри кривої.

*Тест на генерацію точок кривої:*

Перевірити, чи можна згенерувати послідовність точок для кожного рівня кривої. Переконайтесь, що точки правильно розташовані та зв'язані.

*Тест на обробку вхідних даних:*

Впевнитись, що програма обробляє вхідні дані, такі як порядок кривої та інші параметри, та повідомляє про помилки в разі некоректних вхідних даних.

*Тест на відображення кривих Гільберта:*

Переконайтесь, що програма може візуалізувати криві Гільберта відповідно до вказаних параметрів, і відображення вірне.

*Тест на продуктивність:*

Виміряти час виконання генерації кривих різних порядків і переконатись, що програма працює швидко та ефективно.

*Тест на коректність завершення роботи:*

Переконатись, що програма коректно завершує роботу та звільняє всі ресурси після завершення генерації кривої.

Зм.	Аркуш	№ докум.	Підпис	Дата	ДТЕУ 121 02 - 25.МР			
Зав. каф.		Криворучко О.В.		18.10.23	Програмний додаток побудови кривих Гільберта	Стадія	Аркуш	Аркушів
Керівник		Тищенко Д.О.		18.10.23		ПМТ	67	64
Гарант		Котенко Н.О.		18.10.23	Програма та методика тестування	Факультет інформаційних технологій 2м курс, 2 група		
Розробив		Юрченко С.П.		18.10.23				

Загальний план тестування повинен включати як функціональні, так і нефункціональні аспекти програми, щоб переконатися, що вона працює ефективно та надійно



						ДТЕУ 121 02-25.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			68



## ДОДАТКИ

### Додаток А

#### Додаток А «Лістинг коду для побудови кривої Гільберта»

```
import turtle
import tkinter as tk
from tkinter import ttk

# Функція для малювання кривої Гільберта
def hilbert_curve(turtle, order, angle, step):
    if order == 0:
        return
    turtle.right(angle)
    hilbert_curve(turtle, order - 1, -angle, step)
    turtle.forward(step)
    turtle.left(angle)
    hilbert_curve(turtle, order - 1, angle, step)
    turtle.forward(step)
    hilbert_curve(turtle, order - 1, angle, step)
    turtle.left(angle)
    turtle.forward(step)
    hilbert_curve(turtle, order - 1, -angle, step)
    turtle.right(angle)

# Функція для малювання кривої Гільберта зі значеннями,
# введеними користувачем
def draw_hilbert_curve():
    order = int(order_entry.get())
    angle = int(angle_entry.get())
    step = int(step_entry.get())
    # Закриваємо вікно для малювання, якщо воно вже відкрите
    if 'window' in globals():
        window.bye()
    # Створюємо вікно для малювання
    window = turtle.Screen()
```

```

window.bgcolor("white")
# Створюємо черепашку для малювання
t = turtle.Turtle()
t.speed(0) # Найвища швидкість малювання
# Розміщуємо черепашку на правильному місці
t.penup()
t.goto(-300, 0)
t.pendown()
# Малюємо криву Гільберта
hilbert_curve(t, order, angle, step)
# Закриваємо вікно при натисканні на нього
window.exitonclick()
# Створюємо основне вікно користувача
root = tk.Tk()
root.title("Крива Гільберта")
# Створюємо та налаштуємо елементи інтерфейсу
order_label = ttk.Label(root, text="Порядок:")
order_entry = ttk.Entry(root)
angle_label = ttk.Label(root, text="Кут:")
angle_entry = ttk.Entry(root)
step_label = ttk.Label(root, text="Крок:")
step_entry = ttk.Entry(root)
draw_button = ttk.Button(root, text="Малювати",
command=draw_hilbert_curve)
# Розміщуємо елементи інтерфейсу на вікні
order_label.grid(row=0, column=0)
order_entry.grid(row=0, column=1)
angle_label.grid(row=1, column=0)
angle_entry.grid(row=1, column=1)
step_label.grid(row=2, column=0)
step_entry.grid(row=2, column=1)
draw_button.grid(row=3, columnspan=2)
# Запускаємо головний цикл обробки подій
root.mainloop()

```