**Дніпровський ліцей інформаційних технологій**

**при Дніпропетровському національному університеті**

**імені Олеся Гончара**

**Випускна робота**

**на тему:**

**Числові послідовності**

**Виконавець:**

**ліцеїст 11-Г класу**

**Самодрига Олег Андрійович**

**Керівники роботи:**

**Гузєєва Ю.А \_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Боровік Л.І. \_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Дніпро**

**2017**

Зміст

[Вступ 3](#_Toc471562835)

[**Тема роботи** 3](#_Toc471562836)

[**Мета роботи** 3](#_Toc471562837)

[Числові послідовності. Основні поняття про числові послідовності. 3](#_Toc471562838)

[Способи задання числових послідовностей. 5](#_Toc471562839)

[**Властивості числових послідовностей.** 8](#_Toc471562840)

[**Арифметична прогресія та їх властивості. Приклади арифметичних прогресій** 9](#_Toc471562841)

[**Геометрична** **прогресія** 10](#_Toc471562842)

[Cніжинка Коха 11](#_Toc471562843)

[Дракон Хартера-Хейтуэя 14](#_Toc471562844)

[Трикутник Серпінського 15](#_Toc471562845)

[Крива Маньківського 17](#_Toc471562846)

[Килимок Серпинського 17](#_Toc471562847)

[Дерево Піфагора 18](#_Toc471562848)

[Крива Леві 19](#_Toc471562849)

[Фрактал простих чисел 19](#_Toc471562850)

[Теоретична частина з інформатики 21](#_Toc471562851)

[Посібник користувача 26](#_Toc471562852)

[**Початок** 26](#_Toc471562853)

[**Головне меню** 26](#_Toc471562854)

[**Теорія** 27](#_Toc471562855)

[**Практика** 27](#_Toc471562856)

[**Трикутник серпінського** 28](#_Toc471562857)

[**Сніжинка Коха** 29](#_Toc471562858)

[**Квадрат серпінського** 29](#_Toc471562859)

[**Фрактал простих чисел** 30](#_Toc471562860)

[**Геометричні задачі** 31](#_Toc471562861)

[**Геометричні задачі 1-6** 31](#_Toc471562862)

[Структурна схема роботи 32](#_Toc471562863)

[Програмні та апаратні вимоги 32](#_Toc471562864)

[Використані програми 32](#_Toc471562865)

[Висновок 33](#_Toc471562866)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 33](#_Toc471562867)

# Вступ

## **Тема роботи**

Темою моєї кваліфікаційної роботи є Числові послідовності.

Числова послідовність — послідовність дійсних чисел, тобто відображення, яке кожному натуральному числу n ставить у відповідність дійсне число xn. Число xn називають елементом або членом послідовності.

В моїй роботі розглядаються приклади графічних числових послідовностей та їх рішення. Завдяки цим задачам з’являється можливість проводити уроки більш поглиблюючись в тему.

Для вирішення моєї задачі було виконане:

1) Конвертація теоретичної частини в html файл та перенос теорії в проект;

2) для управління проектом був створений інтерфейс за допомогою об’єктів Visual Studio 2010;

3) були розроблені постановки задач та алгоритм їх вирішення;

5) по цьому алгоритму створений проект мовою С#;

6) було проведено тестування програми;

## **Мета роботи**

Мета моєї роботи-поглиблення в вивчанні числових послідовностей. Та створення проекту який би допомагав в проведенні занять.

Данна тема з математики проходиться в дев’ятому класі, тому її застосування завжди актуальна. Але, нажаль, в більшості там зустрічаються задачі по типу: знайти суму значень чи значення шагу числа на n-ому кроці.

# Числові послідовності. Основні поняття про числові послідовності.

Часто у повсякденному житті нам трапляються об’єкти, з якими зручно мати справу, якщо їх попередньо пронумерувати. Наприклад, номери мають місяці і квартали року, дні тижня, під'їзди і квартири будинку, вагони поїзда

Об'єкти, які пронумеровано поспіль натуральними числами 1, 2, 3, .... *п,* ..., утворюють послідовності.

Так, можна казати про послідовність сторінок у книзі, букв у слові, поверхів у будинку тощо.

Об’єкти, які утворюють послідовність, називають членами послідовності. Зрозуміло, що кожний член послідовності має свій номер. Наприклад, січень — це перший член у послідовності місяців року, число 3 — другий член послідовності простих чисел. Узагалі, якщо член послідовності мас номер л, то його називають л-м членом послідовності.

Якщо членами послідовності є числа, то таку послідовність називають числовою.

Наведемо приклади числових послідовностей.

1, 2. З, 4, 5, ... — послідовність натуральних чисел;

2,4, 6, 8, 10, ... — послідовність парних чисел;

0,3; 0,33; 0,333; ... — послідовність десяткових наближень дробу

19, 38, 67, 76, 95 — послідовність двоцифрових чисел, кратних 19;

-1, -2, -3, -4, -5, ... — послідовність від'ємних цілих чисел.

Надалі ми розглядатимемо тільки числові послідовності.

Наведені вище приклади показують, що послідовності бувають скінченнями і нескінченними. Наприклад, послідовність парних натуральних чисел це нескінченна послідовність, а послідовність двоцифрових чисел, кратних 19, — це скінченна послідовність.

Для позначення членів послідовності використовують букви з індексами:

a1, a2, a3, …, an, … .

Індекс указує порядковий номер члена послідовності. Для позначення самої послідовності використовують запис (аn). Наприклад, нехай *(рn) —* послідовність простих чисел. Тоді *p1 =* 2*,**p2 = 3, p3=5, p4 = 7, р5 = 11 і т. д.*

*Послідовність вважають заданою, якщо кожний її член можна визначити за його номером.*

# Способи задання числових послідовностей.

Розглянемо послідовність, у якої перший член дорівнює 1, а кожний наступний член на 3 більший за попередній. Такий спосіб задання послідовності називають описовим. Його можна проілюструвати за допомогою запису а трьома крапками, виписавши кілька перших членів послідовності у порядку зростання номерів:

1, 4, 7, 10, 13, 16, 19 … .

Цей запис доцільно застосовувати тоді, коли зрозуміло, які числа мають бути записані замість трьох крапок.

Наприклад, у послідовності, яку ми розглянемо, зрозуміло, що після числа 19 мас бути записане число 22.

Якщо послідовність с скінченною, то її можна задати за допомогою таблиці. Наприклад, наведена таблиця задає послідовність кубів одноцифрових натуральних чисел:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *п* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| *а*  *«* - | 1 | 8 | 27 | 64 | 126 | 216 | 343 | 512 | 729 |

Послідовності можна задавати за допомогою формул. Наприклад, рівність хn = 2n, де змінна *п* набуває всіх натуральних значень, задає послідовність (xn) натуральних степенів числа два: 2, 4, 8, 16, 32, ... . У таких випадках кажуть, що послідовність задано за допомогою формули n-го члена послідовності.

Розглянемо кілька прикладів.

Формула *аn*= 2n - 1 задає послідовність натуральних непарних чисел:

1, 3, б, 7, 9

Формула yn=(-1)n задає послідовність (yn) у якій усі члени з непарними номерами дорівнюють -1, а з парними номерами дорівнюють 1:

-1,1,-1,1,-1… .

«Формула сn- < задає послідовність (сn), усі члени якої дорівнюють числу 7:

7, 7, 7.,7,7, … .

Послідовність, усі члени якої рівні, називають стаціонарною.

Наведені способи завдання послідовностей допомагають простежити зв’язок між поняттями «функція\* і «послідовність\*.

Розглянемо функцію у = f(x), областю визначення якої е множина натуральних чисел або множина п перших натуральних чисел. Тоді функція f задає нескінченну послідовність f(1), f (2), ..., f (n). ... або скінченну послідовність f(l), f (2), ..., f (n). Іншими словами, нескінченна послідовність являє собою функцію, областю визначення якої с множина N, а скінченна послідовність — це функція, область визначення якої — множина перших n натуральних чисел.

Наприклад, функцію у = х3. D (у) = N. можна розглядати як послідовність квадратів натуральних чисел:

1. 4, 9, 16, 25 … .

Також можна сказати, що. наприклад, стаціонарна послідовність 7, 7, 7, 7,... це відображення множини N на множину {7}.

Нерідко послідовність задають правилом, яке дозволяє знайти наступний член, знаючи попередній.

Розглянемо послідовність (аn), перший член якої дорівнює 1, а кожний наступний член послідовності у 3 рази більший за попередній. Маємо:

1,3,9, 27, 81, … .

Цю послідовність, задану описом, також визначають такі умови:

an=1, an+1=3an

Ці рівності вказують перший член послідовності і правило, користуючись яким, за кожним членом послідовності можна знайти наступний член:

a1 = 1, a2 = За1, = З, а3 = За2, = 9. a4 = За3, = 27

і т. д.

Формулу, яка виражає член послідовності через один або кілька попередніх членів, називають рекурентною формулою У наведеному прикладі це формула аn = Зan-1 . Умови, які визначають перший або кілька перших членів, наливають початковими умовами. У розглядуваному прикладі початкова умова — це а1=1.

Зауважимо, що знання лише однієї рекурентної формули не дозволяє задати послідовність. Мають бути ще вказані початкові умови.

При рекурентному способі задані послідовності перший або кілька перших членів послідовності є заданими, а всі інші обчислюють один за одним. З цієї точки зору спосіб задання послідовності формулою n-го члена видається більш зручним: за його допомогою можна знайти потрібний член послідовності, знаючи лише його номер.

## **Властивості числових послідовностей.**

Числова послідовність - окремий випадок числової функції, тому деякі властивості функцій можна перенести і на послідовності.

Послідовність називається зростаючою, якщо для будь-якого n ∈ N виконується нерівність an<an+1.

Послідовність називається спадною, якщо для будь-якого n ∈ N виконується нерівність an>an+1.

Зростаючі і спадні послідовності називаються монотонними.

1. Послідовність задана формулою , є монотонною, зростаючою, тому що різниця

Приклад:

Тобто an<an+1.

2. Послідовність із спільним членом an=1+(−1)n не є монотонною, тому що a1<a2,a2>a3.

Послідовність називається обмеженою зверху, якщо існує таке число M∈R, що an≤M. При цьому число M називається верхньою межею послідовності.

Послідовність називається обмеженою знизу, якщо існує таке число m∈R, що an≥m. Число m називається нижньою межею послідовності.

Приклад:

1. Послідовність задана формулою an=n; (1,2,3,...,n,...) обмежена знизу, але не обмежена зверху.

2. Послідовність задана формулою an=(−1)n\*n;(−1,2,−3,4,...,(−1)n\*n,...) не обмежена ні зверху, ні знизу.

## **Арифметична прогресія та їх властивості. Приклади арифметичних прогресій**

Арифметичною **прогресією** називають послідовність чисел (членів прогресії) a1,a1+d,a1+2d…  
в якій кожен наступний член відрізняється від попереднього на сталий доданок, який ще **називають кроком або різницею прогресії.** d=an+1-an   
Таким чином, задаючи крок прогресії та її перший член можна знайти будь-який її елемент за формулою.

1) Кожен **член арифметичної прогресії**, починаючи з другого номера **є середнім арифметичним від попереднього та наступного члена** прогресії

Обернене твердження також вірне. Якщо середнє арифметичне сусідніх непарних (парних) членів прогресії рівне члену, який стоїть між ними то дана послідовність чисел є арифметичною прогресією. За цим твердженням дуже просто перевірити будь-яку послідовність.

Також за властивістю арифметичної прогресії, наведену вище формулу можна узагальнити до наступної

В цьому легко переконатися, якщо розписати доданки справа від знака рівності  
Її часто застосовують на практиці для спрощення обчислень в задачах.

2) **Суму *n* перших членів арифметичної прогресії** обчислюють за формулою

Запам'ятайте добре **формулу суми арифметичної прогресії,** вона незамінна при обчисленнях та досить часто зустрічається в простих життєвих ситуаціях.

3) Якщо потрібно знайти не всю **суму**, а частину **послідовності починаючи з *k*-го її члена** то в нагоді Вам стане наступна формула суми

4) Практичний інтерес має **відшукання суми *n* членів арифметичної прогресії починаючи з *k*-го номера**. Для суми використовуйте формулу  
На цьому теоретичний матеріал добігає кінця і переходимо до розв'язування поширених на практиці задач на прогресію.

## **Геометрична** **прогресія**

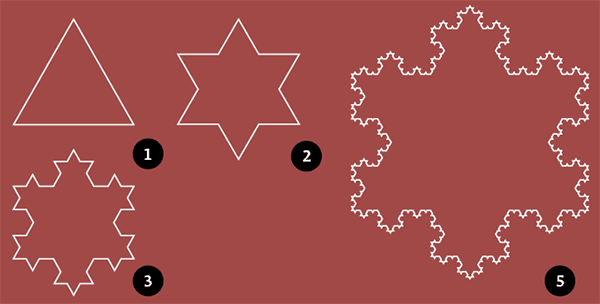
**Геометрична прогресія** не менш важлива в математиці порівняно з арифметичною. **Геометричною прогресією називають** таку послідовність чисел кожен наступний член якої, отримується множенням попереднього на стале число. Конcтанту, яка характеризує швидкість росту або спадання прогресії **називають знаменником геометричної прогресії** і позначають  
Для повного задання геометричної прогресії окрім знаменника необхідна знати або визначити перший її член. Для додатних значень знаменника q>0 прогресія є монотонною послідовністю, причому якщо q<1 то **послідовність чисел є монотонно спадною** і при q>1 **монотонно зростаючою**. Випадок, коли знаменник рівний одиниці q=1 на практиці не розглядається, оскільки маємо послідовність однакових чисел, а їх сумування не важке

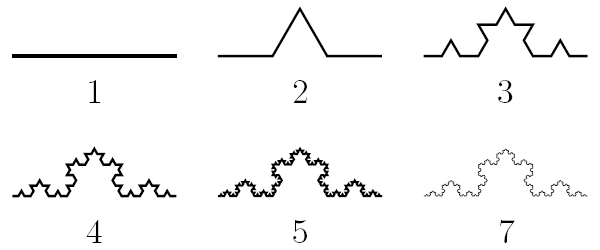
**Загальний член геометричної прогресії** знаходять за формулою

**Суму n перших членів геометричної прогресії** визначають за формулою

Розглянемо розв'язки класичних задач на геометричну прогресію. Почнемо для розуміння теорії з найпростіших.

# Cніжинка Коха

Ця фігура - один з перших досліджених вченими фракталів. Вона виходить з трьох копій кривої Коха, яка вперше з'явилася в статті шведського математика Хельге фон Коха в 1904 році. Ця крива була придумана як приклад безперервної лінії, до якої не можна провести дотичну ні в одній точці. Лінії з такою властивістю були відомі і раніше (Карл Вейерштрасс побудував свій приклад ще в 1872 році), але крива Коха чудова простотою своєї конструкції. Не випадково його стаття називається «Про неперервну криву без дотичних, яка виникає з елементарної геометрії».



Перші етапи побудови кривої Коха

Малюнок і анімація відмінно показують, як по кроках будується крива Коха. Перша ітерація - просто початковий відрізок. Потім він ділиться на три рівні частини, центральна добудовується до правильного трикутника і потім викидається. Виходить друга ітерація - ламана лінія, що складається з чотирьох відрізків. До кожного з них застосовується така ж операція, і виходить четвертий крок побудови. Продовжуючи в тому ж дусі, можна отримувати все нові і нові лінії (всі вони будуть ламаними). А то, що вийде в межі (це вже буде уявний об'єкт), і називається кривою Коха.

Основні властивості кривої Коха

1. Вона неперервна, але ніде не диференційована. Грубо кажучи, саме для цього вона і була придумана - як приклад такого роду математичних «виродків»

2. Має нескінченну довжину. Нехай довжина вихідного відрізка дорівнює 1. На кожному кроці побудови ми замінюємо кожен зі складових лінію відрізків на ламану, яка в 4/3 рази довше. Значить, і довжина всієї ламаної на кожному кроці множиться на 4/3: довжина лінії з номером n дорівнює (4/3)n-1. Тому граничної лінії нічого не залишається, крім як бути нескінченно довгою.

3. Сніжинка Коха обмежує кінцеву площу. І це при тому, що її периметр нескінченний. Це властивість може здатися парадоксальним, але воно очевидно - сніжинка повністю поміщається в коло, тому її площа свідомо обмежена. Площа можна порахувати, і для цього навіть не потрібно особливих знань - формули площі трикутника і суми геометричної прогресії проходять в школі. Для зацікавлених обчислення наведено нижче дрібним шрифтом.

Нехай сторона вихідного правильного трикутника дорівнює a. Тоді його площа . Спочатку сторона дорівнює 1, а площа:. Що відбувається при збільшенні ітерації? Можна вважати, що до вже наявного многоугокутника прилаштовуються маленькі рівносторонні трикутнички. У перший раз їх всього 3, а кожен наступний раз їх в 4 рази більше ніж було в попередній. Тобто на n-му кроці буде добудовано Tn = 3 · 4n-1 трикутників. Довжина сторони кожного з них становить третину від сторони трикутника, добудованого на попередньому кроці. Значить, вона дорівнює (1/3) n. Площі пропорційні квадратах сторін, тому площа кожного трикутнички дорівнює. При великих значеннях n це, до речі, дуже мало. Сумарний внесок цих трикутників в площа сніжинки дорівнює Tn · Sn = 3/4 · (4/9)n · S0. Тому після n-го кроку площа фігури дорівнюватиме сумі S0 + T1 · S1 + T2 · S2 + ... + Tn · Sn =. Сніжинка виходить після нескінченного числа кроків, що відповідає n → ∞. Виходить нескінченна сума, але це сума спадної геометричної прогресії, для неї є формула:Площа сніжинки дорівнює

4. Фрактальна розмірність дорівнює log4 / log3 = log34 ≈ 1,261859 .... Акуратне обчислення потребують чималих зусиль і докладних роз'яснень, тому тут наведено, швидше, ілюстрація визначення фрактальної розмірності. З формули статечної залежності N (δ) ~ (1 / δ) D, де N - число пересічних квадратиків, δ - їх розмір, а D - розмірність, отримуємо, що D = log1 / δN. Це рівність вірно з точністю до додавання константи (однієї і тієї ж для всіх δ). На малюнках зображена п'ята ітерація побудови кривої Коха, зеленим зафарбовані квадратики сітки, які з нею перетинаються. Довжина вихідного відрізка дорівнює 1, тому на верхньому малюнку довжина сторони квадратиків дорівнює 1/9. Зафарбовано 12 квадратиків, log912 ≈ 1,130929 .... Ще не дуже схоже на 1,261859 .... Дивимося далі. На середньому малюнку квадратики в два рази менше, їх розміри 1/18, зафарбовано 30. log1830 ≈ 1,176733 .... Вже краще. Внизу квадратики ще вдвічі менше, зафарбовано вже 72 штуки. log7230 ≈ 1,193426 .... Ще ближче. Далі потрібно збільшувати номер ітерації і одночасно зменшувати квадратики, тоді «емпіричне» значення розмірності кривої Коха буде неухильно наближатися до log34, а в межі і зовсім співпаде.

# Дракон Хартера-Хейтуэя

Вважається, що таку назву фрактал отримав за схожість з традиційними китайськими драконами. По крайній мірі, так здалося вченим, які вперше його досліджували. Кожна ламана-дракон є лише наближенням до дракона-фракталу і складається з відрізків. Ламана з номером n складатиметься з 2n відрізків. Довжина кожного дорівнює 1/2n-ї частини довжини вихідного відрізка. Якщо їх занумерувати числами 0, 1, 2, ... і йти по ламаній, то після кожного відрізка потрібно повертати. Напрямок повороту визначається номером k поточного відрізка:

* повернути праворуч, якщо k дає залишок 1 від ділення на 4;
* повернути ліворуч, якщо k дає залишок 3 від ділення на 4;
* вчинити так само, як після відрізка з номером k / 2, якщо k парне.

Ці правила дозволяють запрограмувати процедуру малювання драконів. Але на практиці використовують те, що чотири криві з номером n - 2 повністю заміщають n-ю криву. Зрозуміло, для цього їх треба масштабувати в 4 рази.

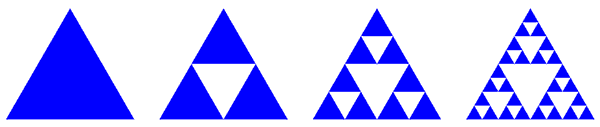
Можна переформулювати ці правила, щоб отримати рекурсивну процедуру побудови ламаних-драконів. На кожному кроці потрібно замінити кожен з відрізків, що складають дану ламану, на куточки - сторони рівнобедреного прямокутного трикутника, у якого цей відрізок є підставою. При цьому потрібно по черзі відкладати ці трикутники то вліво, то вправо по ходу руху від одного кінця ламаної до іншого.

Якщо вирізати кілька плиток в формі фрактала дракона, то їх можна так докласти один до одного, що не залишиться проміжків. Якщо таких плиток багато, то ними можна замостити частину площині. Причому зробити це можна різними способами.

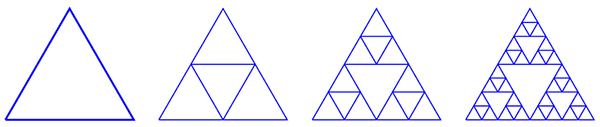
Кілька перших ламаних-драконів можна буквально побудувати своїми руками. Для цього потрібно взяти довгу вузьку і тонку смужку паперу (чим тонше, тим краще!) І скласти її навпіл в одному напрямку кілька разів (зазвичай після 7-8 складань смужка стає дуже товстою, щоб її можна було акуратно скласти ще раз). Після цього потрібно розгорнути смужку. Слідкуйте, щоб в складках виходили прямі кути, тоді в профіль побачите криву дракона.

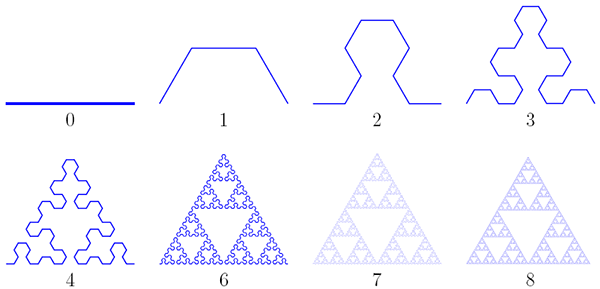
# Трикутник Серпінського

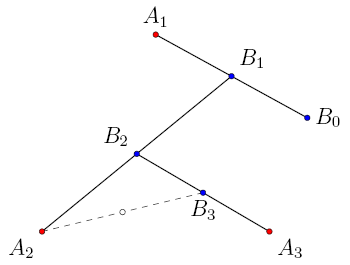
Цей фрактал описав в 1915 році польський математик Вацлав Серпінського. Щоб його отримати, потрібно взяти (рівносторонній) трикутник з начинкою, провести в ньому середні лінії і викинути центральний з чотирьох утворилися маленьких трикутників. Далі ці ж дії потрібно повторити з кожним з решти трьох трикутників, і т. Д. На малюнку показані перші три кроки, а на флеш-демонстрації ви можете потренуватися і отримати кроки аж до десятого.



Побудова трикутника Серпінського

Викидання центральних трикутників - не єдиний спосіб отримати в результаті трикутник Серпінського. Можна рухатися «в зворотному напрямку»: взяти спочатку «порожній» трикутник, потім добудувати в ньому трикутник, утворений середніми лініями, потім в кожному з трьох кутових трикутників зробити те ж саме, і т. Д. Спочатку фігури будуть сильно відрізнятися, але з ростом номера ітерації вони будуть все більше схожими один на одного, а в межі співпадуть.

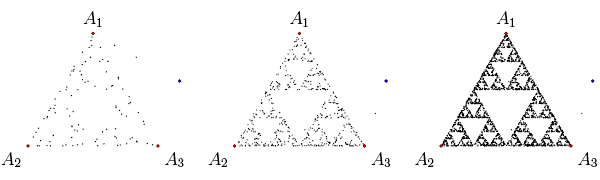
Побудова трикутника Серпінського «в зворотному напрямку»

Наступний спосіб отримати трикутник Серпінського ще більше схожий на звичайну схему побудови геометричних фракталів за допомогою заміни частин чергової ітерації на масштабувати фрагмент. Тут на кожному кроці складові ламану відрізки замінюються на ламану з трьох ланок (вона сама виходить в першій ітерації). Відкладати цю ламану потрібно поперемінно то вправо, то вліво. Видно, що вже восьма ітерація дуже близька до фракталу, і чим далі, тим ближче буде підбиратися до нього лінія.

Гра Хаос

Але і на цьому не все. Виявляється, трикутник Серпінського виходить в результаті однієї з різновидів випадкового блукання точки на площині. Цей спосіб називається «грою Хаос». З його допомогою можна побудувати і деякі інші фрактали.

Суть «гри» така. На площині зафіксовано правильний трикутник A1A2A3. Відзначають будь-яку початкову точку B0. Потім випадковим чином вибирають одну з трьох вершин трикутника і відзначають точку B1 - середину відрізка з кінцями в цій вершині і в B0 (на малюнку праворуч випадково вибралася вершина A1). Те ж саме повторюють з точкою B1, щоб отримати B2. Потім отримують точки B3, B4, і т. Д. Важливо, щоб точка «стрибала» випадковим чином, тобто щоб кожен раз вершина трикутника вибиралася випадково, незалежно від того, що було вибрано в попередні кроки. Дивно, що якщо відзначати точки з послідовності Bi, то незабаром почне проступати трикутник Серпінського. Нижче зображено, що виходить, коли зазначено 100, 500 і 2500 точок.



# Крива Маньківського

Крива Маньківського - класичний геометричний фрактал, запропонований Мінковським. Ініціатором є відрізок, а генератором є ламана з восьми ланок (два рівних ланки продовжують один одного)

* Крива Маньківського ніде не диференційована і не спрямляєма.
* Крива Маньківського не має самоперетинів.
* Крива Маньківського має Хаусдорфорову розмірність (оскільки вона складається з восьми рівних частин, кожна з яких подібна до всієї кривої з коефіцієнтом подібності 1/4). Зокрема,
* Крива Маньківського має нульову міру Лебега.

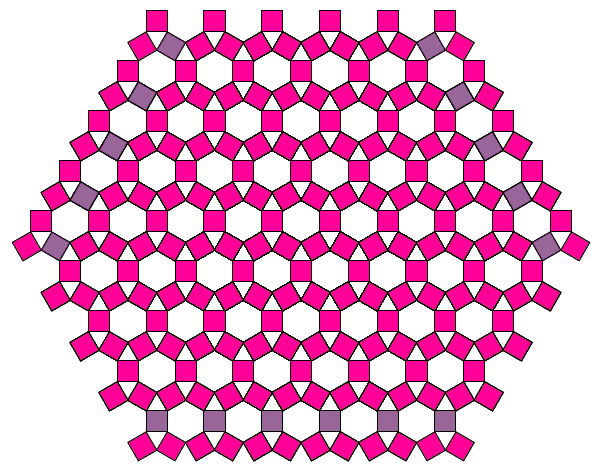
# Килимок Серпінського

Квадратна версія була описана Вацлавом Серпінським в 1916 році. Йому вдалося довести, що будь-яка крива, яку можна намалювати на площині без самоперетинів, гомеоморфна якомусь подмножеству цього дірявого квадрата. Як і трикутник, квадрат можна отримати з різних конструкцій. Справа зображений класичний спосіб: поділ квадрата на 9 частин і викидання центральній частині. Потім той же повторюється для решти 8 квадратів, і т. Д.

# Дерево Піфагора

Називається так тому, що кожна трійка попарно дотичних квадратів обмежує прямокутний трикутник і виходить картинка, якій часто ілюструють теорему Піфагора, «піфагорові штани однієї ширини».

Добре видно, що все дерево обмежена. Якщо найбільший квадрат одиничний, то дерево поміститься в прямокутник 6 × 4. Значить, його площа не перевищує 24. Але з іншого боку, кожен раз додається в два рази більше трійок квадратиків, ніж в попередній, а їх лінійні розміри в √2 раз менше. Тому на кожному кроці додається одна і та ж площа, яка дорівнює площі початковій конфігурації, тобто 2. Здавалося б, тоді площа дерева повинна бути нескінченна! Але насправді протиріччя тут немає, тому що досить швидко квадратики починають перекриватися, і площа приростає не так швидко. Вона все-таки конечна, але, по всій видимості, до сих пір точне значення невідоме, і це відкрита проблема.

Якщо міняти кути при основі трикутника, то будуть виходити трохи інші форми дерева. А при вугіллі 60 ° всі три квадрата виявляться рівними, а дерево перетвориться в періодичний узор на площині: 

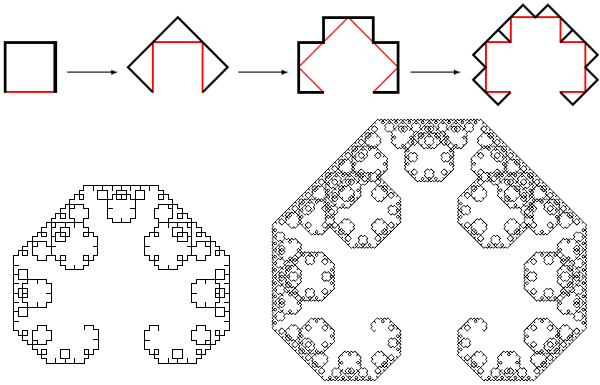
# Крива Леві

Хоча цей об'єкт вивчав ще італієць Ернесто Чезаро в 1906 році, його самоподоба і фрактальні властивості досліджував в 1930-х роках француз Поль П'єр Леві. Фрактальна розмірність межі цього фрактала приблизно дорівнює 1,9340 .... Але це досить складний математичний результат, а точне значення невідоме.

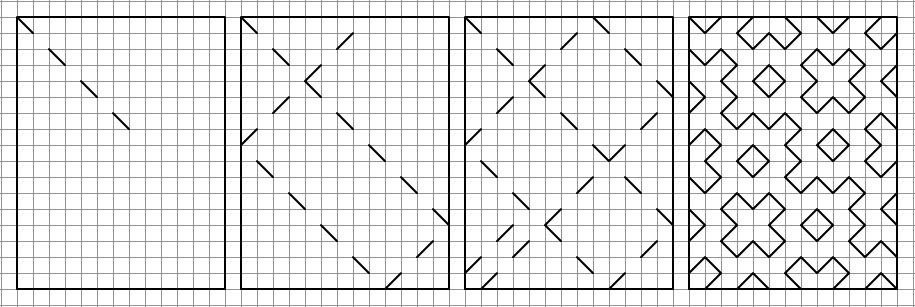
За схожість з буквою «С», написаної хитромудрим шрифтом, її ще називають С-кривої Леві.

Якщо придивитися, то можна помітити, що крива Леві схожа на форму крони дерева Піфагора.

Скособочене крива вийде, якщо замість рівнобедреного прямокутного трикутника на кожному кроці використовувати який-небудь інший прямокутний трикутник. На флеш-демонстрації можна поекспериментувати з різними трикутниками і подивитися, що вийде. Повзунок, можна змінювати співвідношення катетів в трикутнику.

Ще один варіант С-кривої Леві можна побудувати, якщо почати не з відрізка, а з букви П. Нижче показані перші три, восьмий і одинадцятий кроки побудови цієї кривої:

# Фрактал простих чисел

Візьмемо прямокутник зі сторонами q і p. Відправимо промінь (вектор) з кутка в куток. Луч рухається до однієї зі сторін прямокутника, відбивається і продовжує рух до наступної стороні. Це продовжується до тих пір, поки промінь не потрапить в один з решти кутів. Якщо розмір сторони q і p - взаємно прості числа, то виходить візерунок (як ми побачимо пізніше - фрактал). 

# Теоретична частина з інформатики

Було зроблено проект в програмному середовищі C#

В проекті використовувались наступні компоненти програми:

Button – при натисканні виникає подія. Використовується практично у всій програмі для переходу з однієї форми в іншу або для виконання певних дій.

Label – надає елементу управління текст опису яку інформацію під час виконання. Використовується для написання заголовків, текстів чи для виведення результату, наприклад, виведення площини кінцевої фігури.

Textbox – дозволяє користувачеві вводити тексти. Використовується для введення даних о фігурі чи шагу в программі.

WebBrowser – дозволяє користувачеві переглядати веб-сторінки всередині форми. Використовується в розділі "Теоретична частина" для ознайомлення з основною інформацією даної кваліфікаційної роботи.

Picturebox – Показує зображення. Використовується в усьому проекті для виводу зображення

Panel- використовується для виведення інформації в розділі "Практика", "Фрактал простих чисел" для більш доглядного зображення фрактала.

DomainUpDown - дозволяє користувачеві вводити числа Використовується для введення шагу в программі.

TreeView – показує користувачеві ієрархічну колекцію позначених об'єктів, які можуть містити зображення. Використовується в моїй програмі в розділі "Теоретична частина" для відображення заголовків тексту. При натисканні на одного з них, з'являється відповідна веб-сторінка.

Мій проект складається з двох основних розділів: “Теорія”, “Практика”

У розділі “ Теорія ” знаходяться теоретичні відомості про числові послідовності. Як зазначалося вище, в ньому було використані об’єкти: TreeView та WebBrowser. Тобто при виборі заголовку у об’єкті TreeView, – у об’єкта WebBrowser з’являться відповідний розділ документу. Наприклад:private void frmTeor\_Load(object sender, EventArgs e)

{

string path = Application.StartupPath;

webBrowser1.Navigate(path + "\\texts\\1.htm");

}

private void treeView1\_AfterSelect(object sender, TreeViewEventArgs e)

{

string path = Application.StartupPath;

string st1 = treeView1.SelectedNode.Text;

int sqq = treeView1.SelectedNode.Index;

if (st1 == "Числові послідовності. Основні поняття про числові послідовності")

webBrowser1.Navigate(path + "\\texts\\1.htm");

}

Розділ “Практика” є основним і найбільш складним. Тут знаходяться найбільш цікаві елементи коду програми.

Розглянемо перший підрозділ "Трикутник Серпінського".

Користувач самостійно вводить шаг.

double kolvo\_ost, kolvo\_vil;

w = pictureBox1.Width;

h = pictureBox1.Height;

n = int.Parse(Microsoft.VisualBasic.Interaction.InputBox("Введіть номер шагу.(Число повинно бути натуральним)"));

while (n < 1)

{

n = int.Parse(Microsoft.VisualBasic.Interaction.InputBox("Введіть номер шагу.(Число повинно бути натуральним)"));

}

Після чого програма підраховує кількість вилучених фігур та кількість залишившихся

kolvo\_vil = 1 \* (Math.Pow(3, n) - 1) / 2;

label5.Text = Convert.ToString("Кількість вилучених трикутнриків " + kolvo\_vil);

kolvo\_ost = Math.Pow(3, n);

label6.Text = Convert.ToString("Кількість залишившихся трикутнриків " + kolvo\_ost);

Останнім кроком стає виведення зображення отриманої фігури

private void DrawTriangle(int level, PointF top, PointF left, PointF right)

{

if (level == 0)

{

PointF[] points = new PointF[3]

{

top, right, left

};

g.FillPolygon(Brushes.BlueViolet, points);

}

else

{

var leftMid = MidPoint(top, left);

var rightMid = MidPoint(top, right);

var topMid = MidPoint(left, right);

DrawTriangle(level - 1, top, leftMid, rightMid);

DrawTriangle(level - 1, leftMid, left, topMid);

DrawTriangle(level - 1, rightMid, topMid, right);

}

}

private PointF MidPoint(PointF p1, PointF p2)

{

return new PointF((p1.X + p2.X) / 2f, (p1.Y + p2.Y) / 2f);

}

private void button8\_Click(object sender, EventArgs e)

{

frmPract thirt\_form = new frmPract();

thirt\_form.StartPosition = FormStartPosition.CenterScreen;

this.Hide();

thirt\_form.ShowDialog();

}

Розглянемо другий підрозділ "Сніжинка Коха".

Користувач самостійно вводить шаг та розмір сторони.

n = int.Parse(Microsoft.VisualBasic.Interaction.InputBox("Введіть номер шагу.(Число повинно бути натуральним)"));

if (n < 1)

{n = int.Parse(Microsoft.VisualBasic.Interaction.InputBox("Введіть номер шагу.(Число повинно бути натуральним)"));

}

l = Double.Parse(Microsoft.VisualBasic.Interaction.InputBox("Введіть довжину сторони"));

Після чого програма підраховує плошину та периметр першої фігури та сумму отриманих площин, кількість фігур на n-ому кроці.

s0 = (Math.Sqrt(3) / 4) \* l;

label1.Text = Convert.ToString("Стартова площина " + Math.Round(s0, 2));

s = s0\*8/5;

label2.Text = Convert.ToString("Площина фігури " + Math.Round(s, 2));

T = (3 \* Math.Pow(3, n1 - 2))+1;

label3.Text = Convert.ToString("Кількість фігур на n-ому кроці " + T);

if (n < 2)

{

label2.Text = Convert.ToString("Площина фігури " + s0);

label3.Text = Convert.ToString("Кількість фігур на n-ому кроці " + 1);

}

Останнім кроком стає виведення зображення отриманої фігури

Pen pen1 = new Pen(Color.Green, 1);

if (iter > 0)

{

var p4 = new PointF((p2.X + 2 \* p1.X) / 3, (p2.Y + 2 \* p1.Y) / 3);

var p5 = new PointF((2 \* p2.X + p1.X) / 3, (p1.Y + 2 \* p2.Y) / 3);

var ps = new PointF((p2.X + p1.X) / 2, (p2.Y + p1.Y) / 2);

var pn = new PointF((4 \* ps.X - p3.X) / 3, (4 \* ps.Y - p3.Y) / 3);

g.DrawLine(pen1, p4, pn);

g.DrawLine(pen1, p5, pn);

g.DrawLine(pen2, p4, p5);

Thread.Sleep(100);

Draw(p4, pn, p5, iter - 1);

Draw(pn, p5, p4, iter - 1);

Draw(p1, p4, new PointF((2 \* p1.X + p3.X) / 3, (2 \* p1.Y + p3.Y) / 3), iter - 1);

Draw(p5, p2, new PointF((2 \* p2.X + p3.X) / 3, (2 \* p2.Y + p3.Y) / 3), iter - 1);

}

return iter;

Розглянемо третій підрозділ "квадрат Серпінського".

Користувач самостійно вводить шаг.

n = int.Parse(Microsoft.VisualBasic.Interaction.InputBox("Введіть номер шагу.(Число повинно бути натуральним)"));

while (n < 1)

{

n = int.Parse(Microsoft.VisualBasic.Interaction.InputBox("Введіть номер шагу.(Число повинно бути натуральним)"));}

Після чого програма підраховує кількість вилучених фігур та кількість залишившихся

kolvo\_vil = (Math.Pow(8, n) - 1) / 7;

kolvo\_ost = Math.Pow(8, n);

label5.Text = Convert.ToString("Кількість вилучених квадратів " + kolvo\_vil);

label6.Text = Convert.ToString("Кількість залишившихся квадратів " + kolvo\_ost);

Останнім кроком стає виведення зображення отриманої фігури

{

if (n == 0)

{

g.FillRectangle(Brushes.OrangeRed, carpet);

}

else

{

var width = carpet.Width / 3f;

var height = carpet.Height / 3f;

var x1 = carpet.Left;

var x2 = x1 + width;

var x3 = x1 + 2f \* width;

var y1 = carpet.Top;

var y2 = y1 + height;

var y3 = y1 + 2f \* height;

DrawCarpet(n - 1, new RectangleF(x1, y1, width, height));

DrawCarpet(n - 1, new RectangleF(x2, y1, width, height));

DrawCarpet(n - 1, new RectangleF(x3, y1, width, height));

DrawCarpet(n - 1, new RectangleF(x1, y2, width, height));

DrawCarpet(n - 1, new RectangleF(x3, y2, width, height));

DrawCarpet(n - 1, new RectangleF(x1, y3, width, height));

DrawCarpet(n - 1, new RectangleF(x2, y3, width, height));

DrawCarpet(n - 1, new RectangleF(x3, y3, width, height));

}

pictureBox1.BackgroundImage = serp;

Розглянемо четвертий підрозділ "Фракттал простих чисел".

Користувач самостійно вводить розмір лінії по x та по y та вибирає починати з оступу чи ні.

gf.Draw(size, (int)xStepNumericUpDown.Value, (int)yStepNumericUpDown.Value, checkBox1.Checked);

if (offset)

lineStart.Offset(dx, dy);

if ((lineStart.X >= bottomRight.X) || (lineStart.Y >= bottomRight.Y)) return;

g.DrawLine(Pen, lineStart.X, lineStart.Y, lineStart.X + dx, lineStart.Y + dy);

lineStart.Offset(dx, dy);

Після чого починається відмальовування

g.DrawLine(Pen, lineStart.X, lineStart.Y, lineStart.X + dx, lineStart.Y + dy);

lineStart.Offset(dx, dy);

do

{

if (lineStart.X == start.X || lineStart.X == bottomRight.X)

{

dx \*= -1;

}

if (lineStart.Y == start.Y || lineStart.Y == bottomRight.Y)

{

dy \*= -1;

}

offset = !offset;

if (offset)

{

g.DrawLine(Pen, lineStart.X, lineStart.Y, lineStart.X + dx, lineStart.Y + dy);

}

lineStart.Offset(dx, dy);

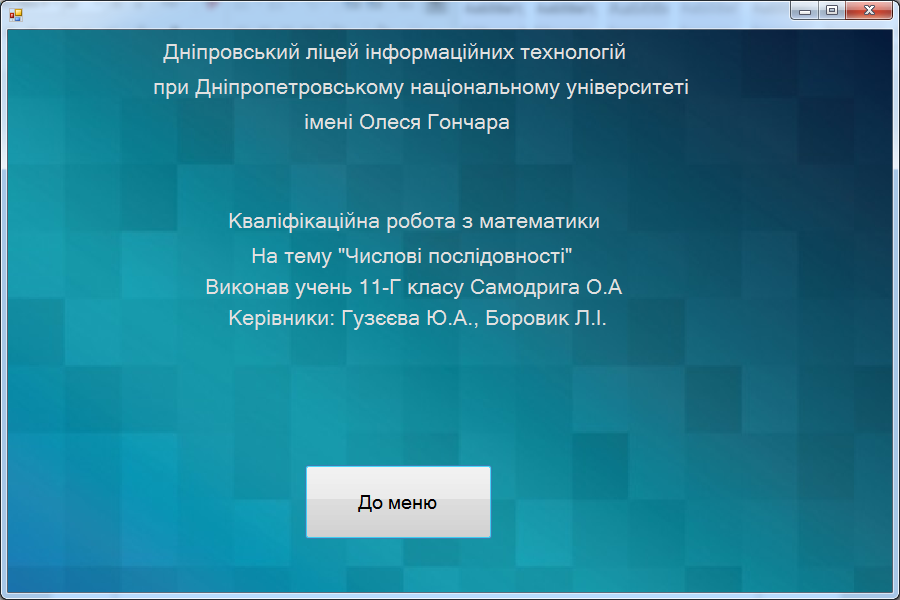
} while (!(lineStart == start || lineStart == bottomRight || lineStart == bottomLeft || lineStart == topRight));

Розглянемо п’ятий підрозділ "Геометричні задачі".

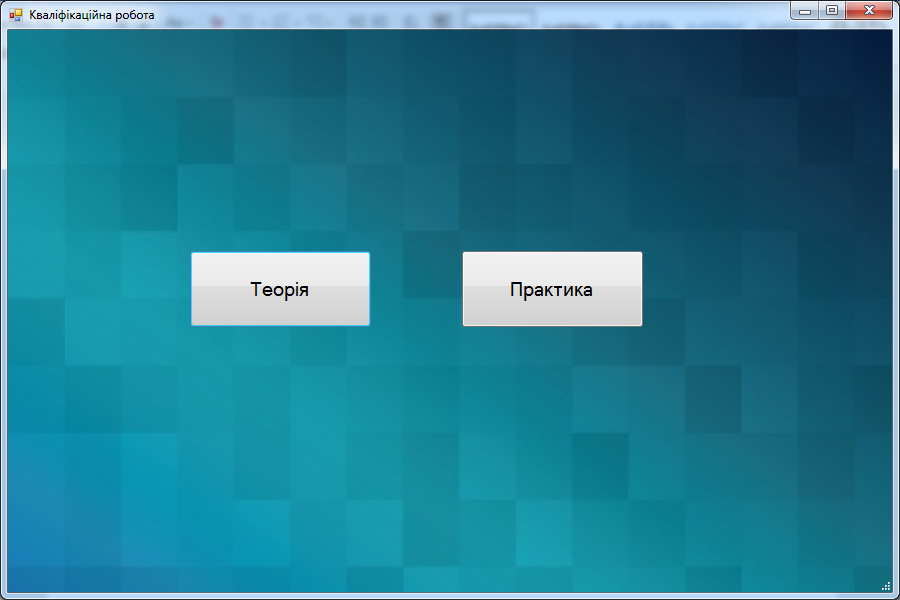
Сам розділ поділяється на дев’ять задач в яких можна ввести данні зв’язані з фігурою (радіус, довжину і т.д)

# Посібник користувача

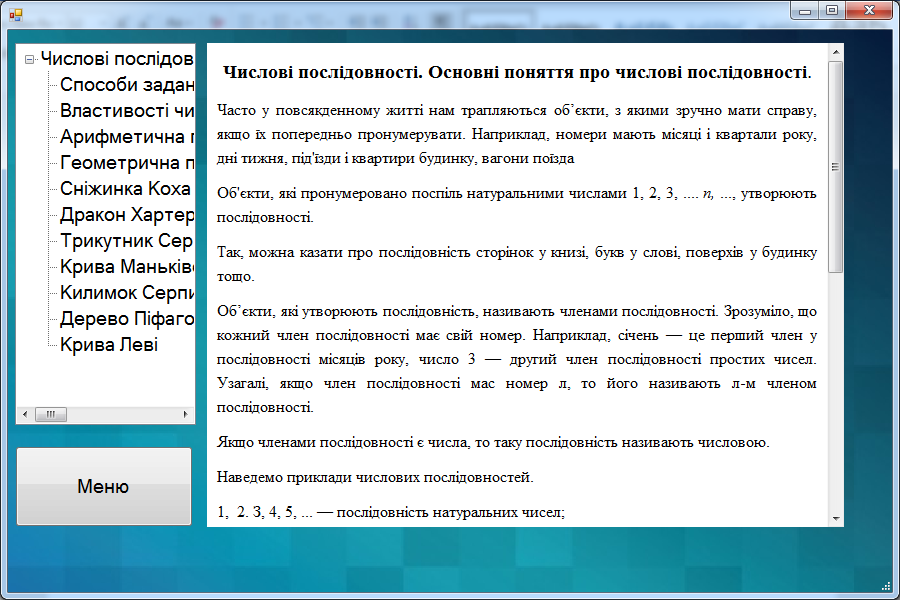
## **Початок**

При запуску програми з'являється форма. Щоб потрапити на наступну форму, потрібно клікнути кнопку "До меню".

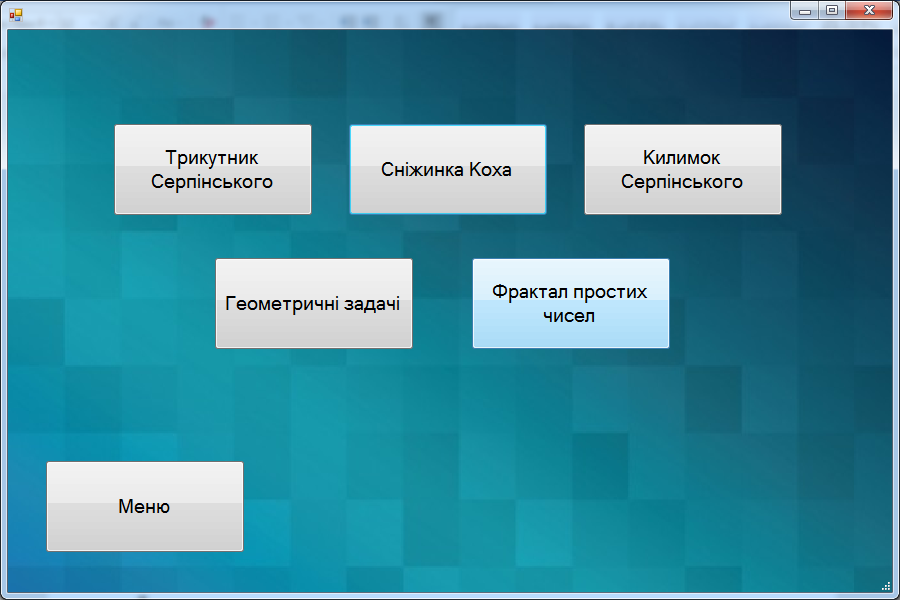
## **Головне меню**

З'являється наступна форма, де кожен з пунктів меню переведе вас до різних розділів програми: "Теорія" та "Практика".

## **Теорія**

Розділ "Теорія" містить теоретичні відомості про числові послідовності. Щоб розглянути вибраний вами розділ, треба натиснути на відповідну назву розділу у верхньому листі. Щоб Для виходу з розділу "Теорія" треба натиснути на кнопку "Меню".

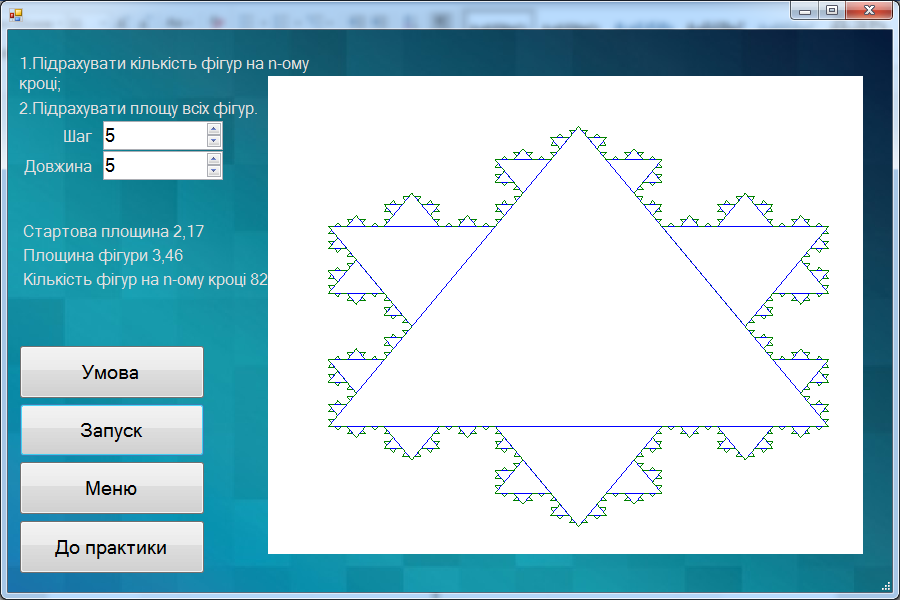
## **Практика**

В розділі практика ви можете ознайомитися з основними завданнями та перейти до них.

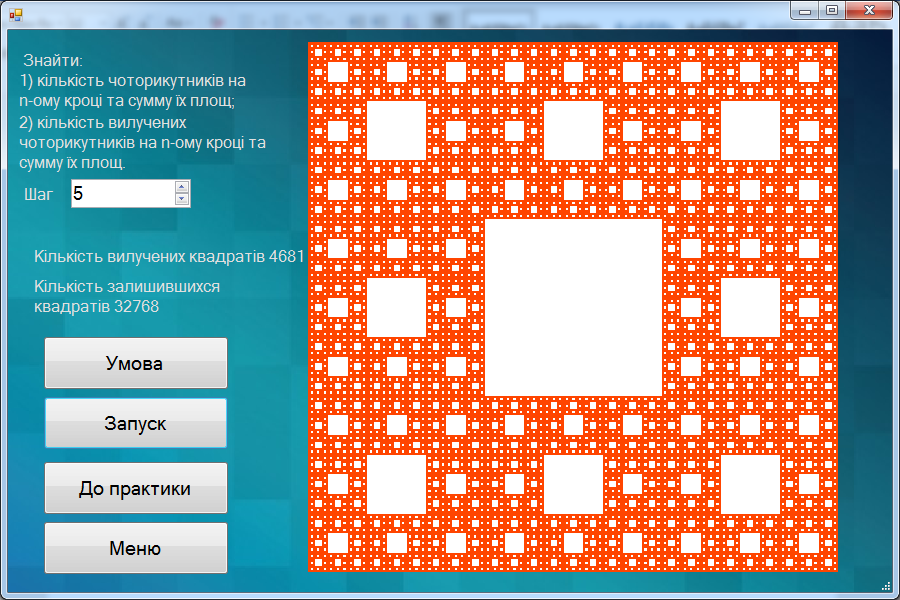
## **Трикутник Серпінського**

В даній задач потрібно ввести шаг. Можна спостерігати малюнок задачі та розрахунки. У відповідній вкладці можна подивитися умову задачі.

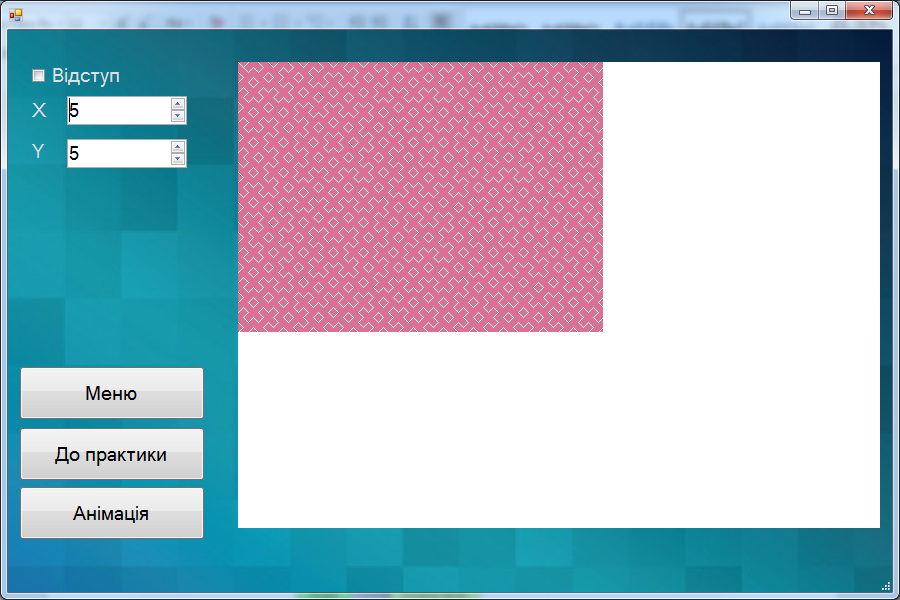
## **Сніжинка Коха**

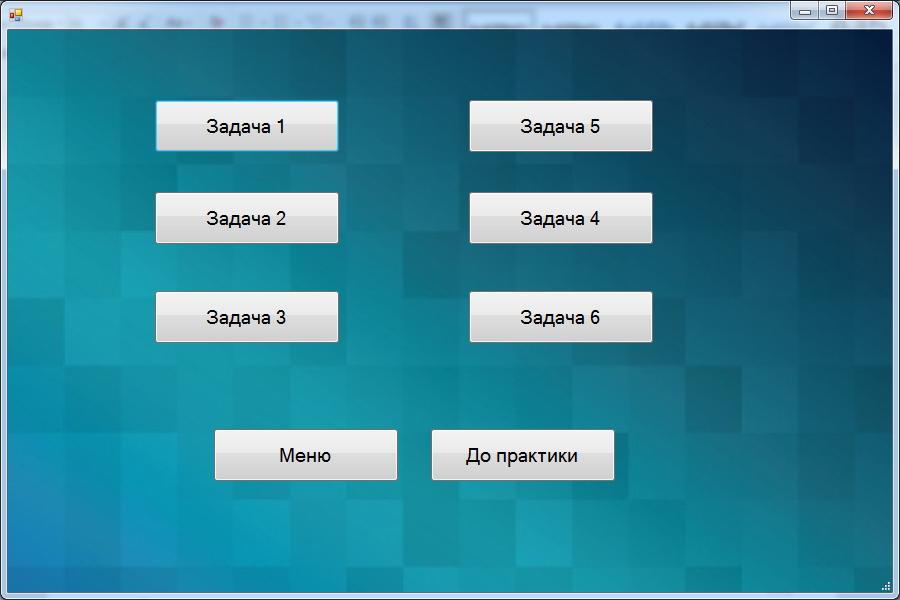
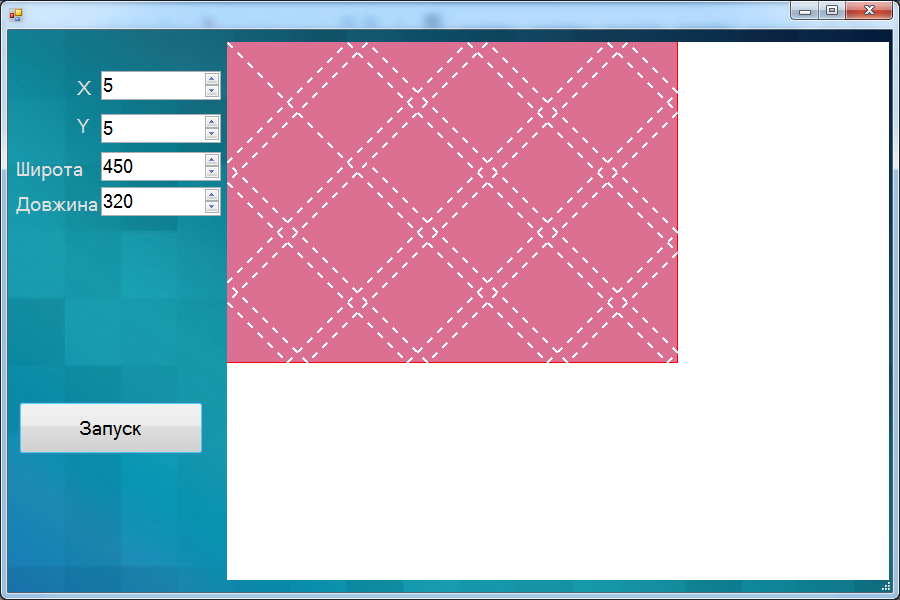
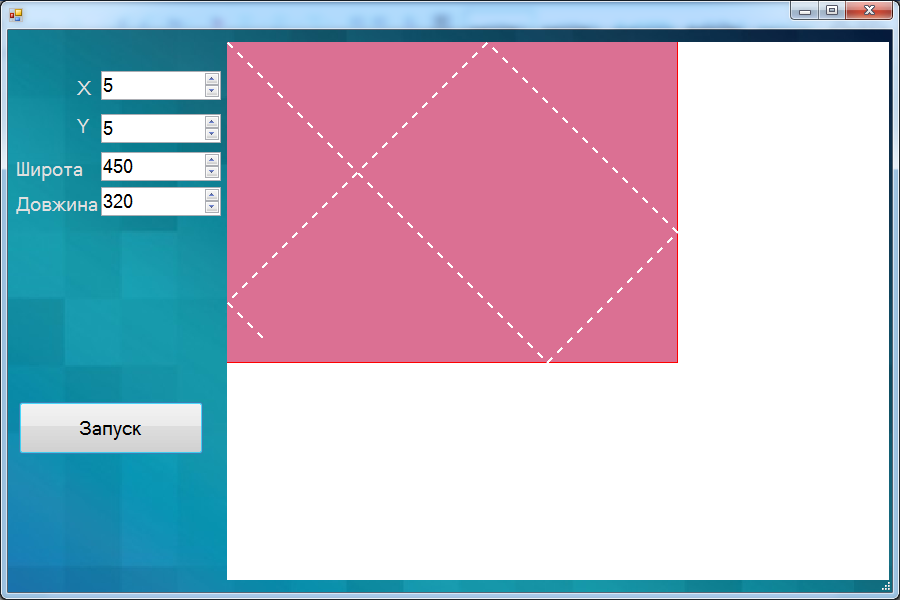
****В даній задачі потрібно ввести шаг та довжину. Можна спостерігати малюнок задачі відповідно до шагу та розрахунки пов’язанні з довжиною та шагом.

## **Квадрат серпінського**

В даній задач потрібно ввести шаг. Можна спостерігати малюнок задачі та разрахунки.

## **Фрактал простих чисел**

В даній програмі вводяться координати лінії по x та y від цього залежить зміщення. При русі миші по білій поверхні змінюється розмір квадрата в котрому знаходяться ці лініх.

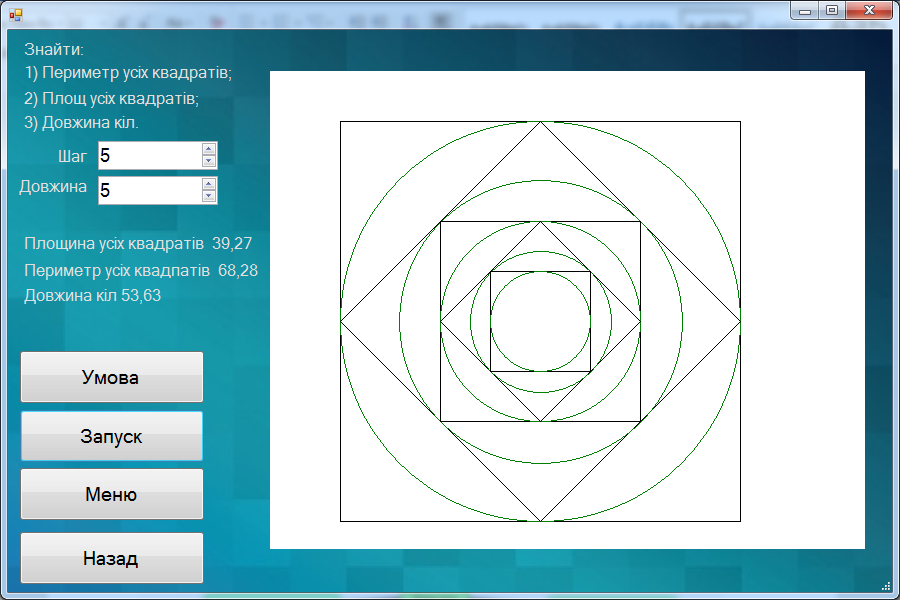
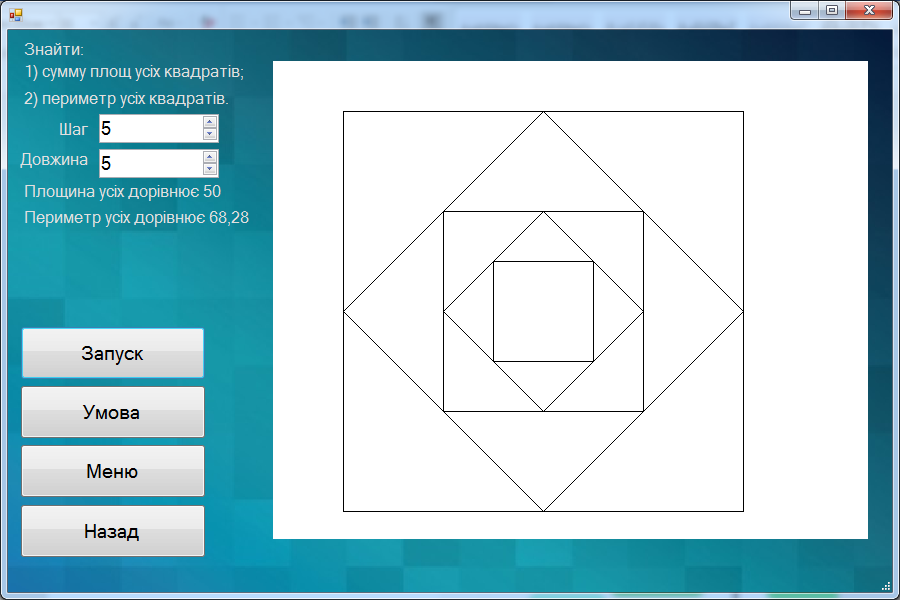
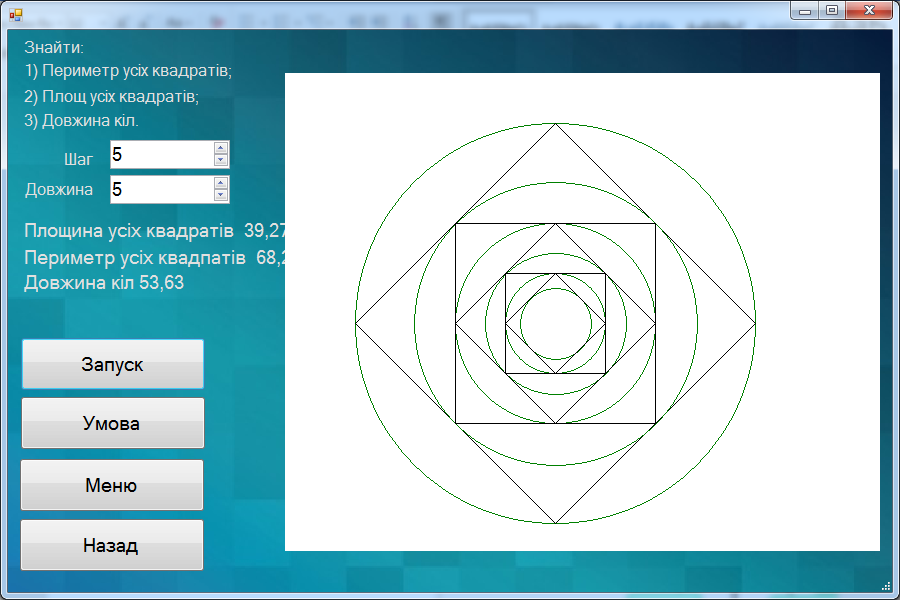
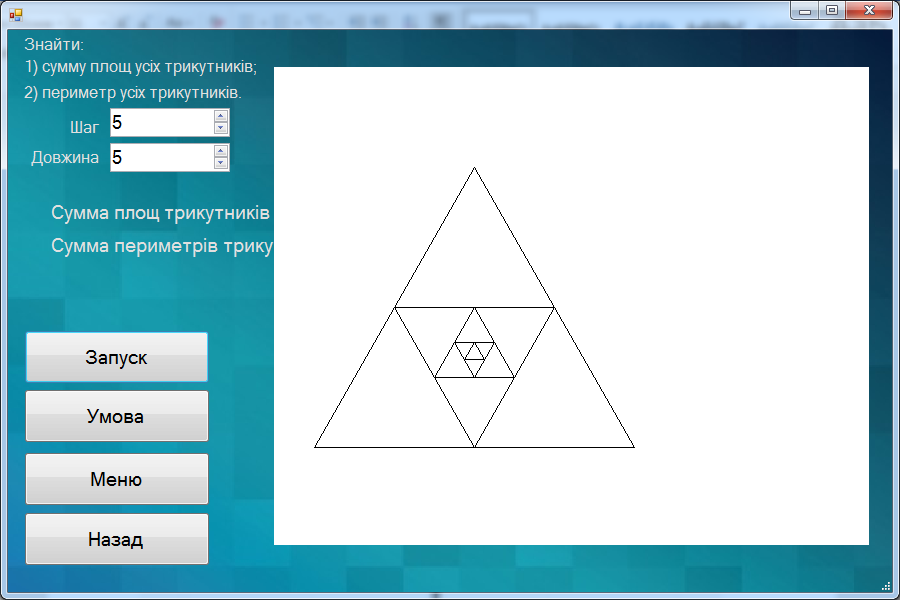
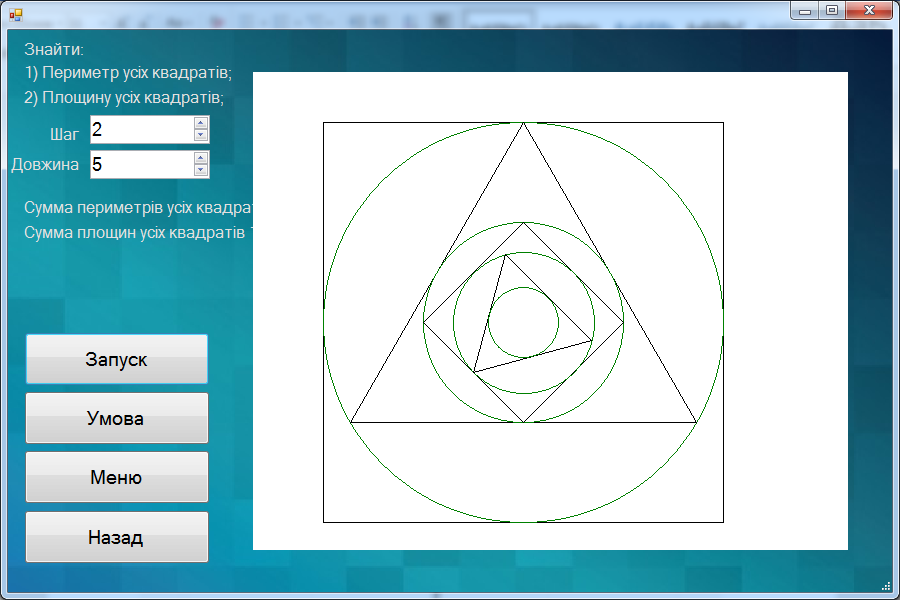
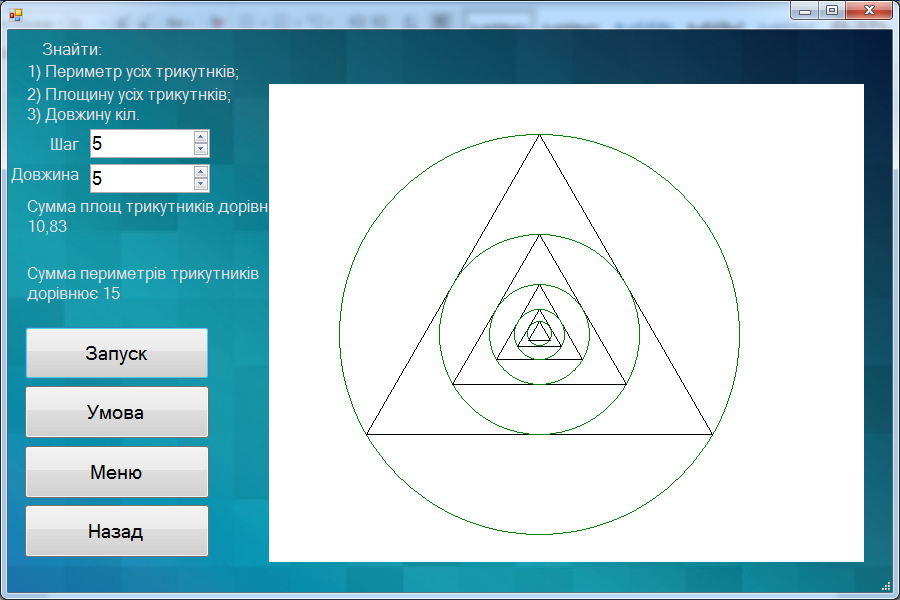
 По нажаті кнопки анімація з’являється анімірований варіант цієї програми. Вводиться розмір лінії та розмір поверхні. При нажаті на кнопку «Запуск» починається анімація

## **Геометричні задачі**

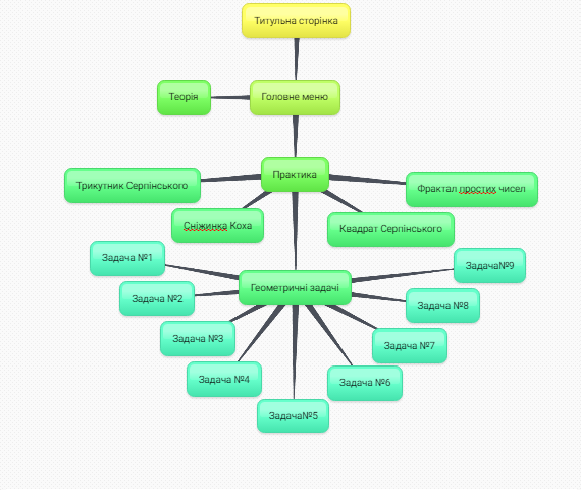
При нажаті на кнопку геометричні задачі з’являєтеся перелік з геометричних задач, при нажаті на які здійсняється перехід.

## **Геометричні задачі 1-6**

При нажаті на номер задачі, можна ознайомитися з шестю подібними задачами, де можна спостерігати аніміровану побудову та задачі та відповідні розрахунки.



# Структурна схема роботи



# Програмні та апаратні вимоги

* ОС Windows 7/8/8.1/10 з архітектурою 32 розряди(x86) або 64 розряди (х64).
* Процесор з частотою 1.6 ГГц або більш потужний.
* ОЗУ об'ємом 25 мб.
* Дозвіл екрану не менше 1280х772.
* Заінстальовані: середа розробки Visual Studio 2010.

# Використані програми

* Microsoft Visual Studio 2010
* Microsoft Word 2010
* Microsoft Visual Studio 2013

**Комплектація програми**

* Меню.exe – файл запуску програми;
* Файли .png – графічні об’єкти, використані у програмі;
* Файли .htm – текстові файли з теорією;

# Висновок

В ході даної роботи була створена програма, яка допоможе у вивчені та поглиблені знань з тем «Числові послідовності».

Данна тема є дуже актуальною оскільки роздивляється тему яка вивчається в 9 класі середньо освітнього навчального закладу.

Створена програма допоможе у вивченні теорії, розширить знання з данної тематики. Її також можна використати в курсі вивчення та проведенні занять 9 класа.

У ході створення даної роботи було детальніше вивчено теорію, розглянуто більш ускладнені задачі, а також опрацьовано в середовищі С#.

В процесі виконання задачі було зроблено багато роботи: для управління проектом був створений інтерфейс за допомогою об’єктів VisualBasic 2010; були розроблені математична постановка задачі та покроковий алгоритм вирішення задачі; по цьому алгоритму створений проект за допомогою VisualBasic 2010.

В ході цієї роботи були розширенні поглибленні знання з даної тематики та програмування.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Книги:

Мерзляк, Полонський, Якір – "Алгебра поглиблене вивчення 9 клас"