

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра інженерії програмного забезпечення

**КУРСОВИЙ ПРОЕКТ**  
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

з дисципліни: «Об'єктно-орієнтоване програмування»  
на тему:  
«Розробка програми побудови фракталів»

студента I курсу групи ПЗ-20-4  
спеціальності 121 «Інженерія програмного  
забезпечення»

Кормиша Романа Івановича

(прізвище, ім'я та по-батькові)

Керівник: Власенко О.В

\_\_\_\_\_.

Дата захисту: " \_\_\_\_ " \_\_\_\_\_ 20\_\_ р.

Національна шкала \_\_\_\_\_

Кількість балів: \_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_

Члени комісії

\_\_\_\_\_.

(підпис)

(прізвище та ініціали)

\_\_\_\_\_.

(підпис)

(прізвище та ініціали)

\_\_\_\_\_.

(підпис)

(прізвище та ініціали)

Житомир – 2021

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»  
Факультет інформаційно-комп'ютерних технологій  
Кафедра інженерії програмного забезпечення  
Освітній рівень: бакалавр  
Спеціальність 121 «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»  
в.о Зав. кафедри  
\_\_\_\_\_ А.В.Морозов  
“ \_\_\_\_ ” \_\_\_\_\_ 20\_\_р.

ЗАВДАННЯ  
НА КУРСОВИЙ ПРОЕКТ СТУДЕНТУ

Кормишу Роману Івановичу

- Тема роботи: Розробка програми побудови фракталів.  
керівник роботи: Власенко О.В.  
Строк подання студентом: “ \_\_\_\_\_ ” травня \_\_\_\_ 20\_\_р.
- Вихідні дані до роботи: Розробити програму побудови фракталів, яка дозволить будувати фрактали відповідно до введених користувачем даних та зберігати їх як зображення.
- Зміст розрахунково-пояснювальної записки(перелік питань. Які підлягають розробці)
  - Постановка завдання
  - Аналіз аналогічних розробок
  - Алгоритми роботи програми
  - Опис роботи програми
  - Програмне дослідження
- Перелік графічного матеріалу(з точним зазначенням обов'язкових креслень)
  - Презентація до КП
  - Посилання на репозиторій: <https://gitlab.com/2020-2024/ipz-20-4/kormysh-roman/coursework.git>
- Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посади консультанта	Підпис, дата	
		завдання видав	завдання прийняв


6. Дата видачі завдання “ 17 ” лютого \_\_\_\_\_ 2021 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів курсового проекту	Строк виконання етапів проекту	Примітки
1	Постановка задачі	17.02.21 – 28.02.21	Виконано
2	Пошук, огляд та аналіз аналогічних розробок	28.02.21 – 01.03.21	Виконано
3	Формулювання технічного завдання	01.02.21 – 04.03.21	Виконано
4	Опрацювання літературних джерел	04.02.21 – 05.03.21	Виконано
5	Проектування структури	05.02.21 – 11.03.21	Виконано
6	Написання програмного коду	12.03.21 – 26.03.21	Виконано
7	Відлагодження	26.03.21 – 21.05.21	Виконано
8	Написання пояснювальної записки	12.04.21 – 21.05.21	Виконано
9	Захист	01.06.21	

**Студент**

\_\_\_\_\_  
(підпис)

**Кормиш Р.І.**

(прізвище та ініціали)

**Керівник проекту**

\_\_\_\_\_  
(підпис)

**Власенко О.В.**

(прізвище та ініціали)

## РЕФЕРАТ

Пояснювальна записка до курсового проекту на тему «Програма побудови фракталів» складається з переліку умовних скорочень, вступу, трьох розділів, висновків, списку використаної літератури та додатків.

Текстова частина викладена на 40 сторінках друкованого тексту.

Пояснювальна записка має 34 сторінки додатків. Список використаних джерел містить 10 найменувань і займає 1 сторінку. В роботі наведено 25 рисунків. Загальний обсяг роботи – 74 сторінок.

У першому розділі було обґрунтовано створення програми для створення фракталів та збереження їх зображень в різних форматах.

У другому розділі проведено проектування і розробка програмного продукту.

У третьому розділі проведено тестування.

Висновок містить в собі результати виконаної роботи створення програми для побудови, дослідження та збереження фракталів.

У додатку представлений лістинг розробленого програмного продукту.

Ключові слова: ООП, ФРАКТАЛИ, МНОЖИНА МАНДЕЛЬБРОТА, КРИВА ХАРТЕРА - ХЕЙТУЕЯ, ПАПОРОТНИК БАРНСЛІ, ФРАКТАЛЬНЕ ДЕРЕВО.

					ДУ «Житомирська політехніка».21.121.12.000 - ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.	Кормиш Р.І.				Розробка програма побудови фракталів	Літ.	Арк.	Аркушів
Керівник	Власенко О.В						4	74
Н. контр.	.							
Затвердив	Морозов А.В					ФІКТ Гр. ІПЗ-20-4		

<b>Зміст</b>	
<b>ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ .....</b>	<b>6</b>
<b>ВСТУП .....</b>	<b>7</b>
<b>РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ.....</b>	<b>8</b>
<b>1.1 Аналіз задачі, засобів та методів її вирішення.....</b>	<b>8</b>
<b>1.2 Аналіз існуючого програмного забезпечення за тематикою курсової роботи.....</b>	<b>9</b>
<b>Висновки до першого розділу :.....</b>	<b>14</b>
<b>РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....</b>	<b>15</b>
<b>2.1 Проектування загального алгоритму роботи програми .....</b>	<b>15</b>
Загальна схема роботи програми :.....	17
Схема множини Мандельброта: .....	18
Схема фрактального дерева: .....	19
Схема папоротника Барнслі:.....	20
Схема кривої Хартера – Хейтуея:.....	21
<b>2.2 Розробка функціональних алгоритмів роботи програми .....</b>	<b>22</b>
Діаграма класів та компонентів :.....	23
<b>2.3 Розробка програмного забезпечення .....</b>	<b>24</b>
<b>Висновки до 2 розділу : .....</b>	<b>29</b>
<b>РОЗДІЛ 3 ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ .....</b>	<b>30</b>
<b>3.1 Опис роботи з програмним додатком (Опис інтерфейсу).....</b>	<b>30</b>
<b>3.2 Тестування роботи програмного забезпечення .....</b>	<b>36</b>
<b>Висновки до 3 розділу : .....</b>	<b>38</b>
<b>ВИСНОВКИ .....</b>	<b>39</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>40</b>
<b>ДОДАТКИ .....</b>	<b>41</b>

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		5

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

VS – Visual Studio

ПЗ – Програмне забезпечення

ППФ – Програма побудови фракталів

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				6
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВСТУП

У цій курсовій роботі буде наведено процес створення програми побудови фракталів.

Програми побудови фракталів — це комп'ютерна програма, яка будує фрактали, опираючись на фрактальні формули. Подібні програми призначені для візуалізації фрактальних формул, створенню фрактальної графіки, детального огляду створених фракталів та їх збереження. Тому в роботі ППФ можна виділити два основних етапи: перший — побудова фракталу за формулою, другий — робота зі створеним фракталом та його збереження.

ППФ широко застосовуються в науці. Найбільш корисним застосуванням фракталів є у комп'ютерній науці - фрактальне стиснення даних. Фрактали використовуються в нафтохімії при моделюванні пористих матеріалів, в біології для опису внутрішніх органів, в медицині для відображення биття серця, у фізиці при моделюванні нелінійних процесів. В природі фрактальними властивостями володіють безліч об'єктів - кора дерев, хмари, сніжинки тощо. Фрактали використовують в комп'ютерних іграх, де рельєфна місцевість, зазвичай, є фрактальним зображенням. Слідуючи з цих тверджень, фрактали та фрактальна графіка потрібні усюди, тому створення ППФ є актуальним в наш час.

**Метою створення** даного курсового проекту є розробка програми для побудови фракталів з додатковими можливостями для їх дослідження.

**Об'єктом дослідження** є технології побудови фракталів.

**Предмет дослідження** — є вивчення можливостей інтерфейсу програмування додатків Windows Forms (WF) та використання їх для побудови фракталів, їх дослідження та збереження.

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				7
Змн.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ

### 1.1 Аналіз задачі, засобів та методів її вирішення

Задача полягає в тому, що потрібно:

- 1) Визначити потреби звичайного користувача
- 2) Вияснити способи реалізації відмальовування фракталів на формі
- 3) Зрозуміти алгоритми побудови фракталів .
- 4) Продумати можливість приближення, віддалення , повернення до початку та управління програмою за допомогою “hot keys”.
- 5) Налагодити можливість збереження фракталу як зображення в різних форматах.
- 6) Передбачити можливість задання параметрів фракталу користувачем.
- 7) Реалізувати адаптивність форм.
- 8) Продумати дизайн та оформлення програми.
- 9) Реалізувати візуалізацію процесу побудови фракталу.

В процесі створення ППФ бралися різні аспекти проекту, розроблялася найпростіша версія, а після того вдосконалювалася, чи повністю перероблювалася.

Для реалізації даного програмного продукту чудово підходить інтегроване середовище розробки програмного забезпечення Visual Studio, а саме платформа Windows Forms (WF), адже вона має велику кількість елементів управління та забезпечує високий рівень стабільності роботи програми. Також слід обрати платформу .Net Framework і мову програмування C#.

При реалізації треба використовувати об'єктно-орієнтований підхід, тому що тоді завдання спрощується і з'являється можливість оперувати більшими по об'єму та складнішими програмами. Також це допомагає уникнути деяких помилок. Також у подальшому, з легкістю, можна використовувати даний проект в іншому, або переробити при необхідності.

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				8
Змн.	Арк.	№ докум.	Підпис	Дата		



За рахунок використання класів, можна легко модифікувати існуючі елементи без зміни вже готових.

## 1.2 Аналіз існуючого програмного забезпечення за тематикою курсової роботи.

При аналізі вже існуючого ПЗ за тематикою курсової роботи було виявлено декілька проектів. Всі вони схожі один на одного і мають схожу реалізацію, адже всі вони розроблені для побудови фракталів та створення фрактальної графіки.

Аналогічні програмні додатки за заданою темою:

### Ultra Fractal.

Розробник: Frederik Slijkerman.

Мова програмування: Delphi.

Посилання на ПО: <https://www.ultrafractal.com/>

Розглянемо перший аналог програму Ultra Fractal:

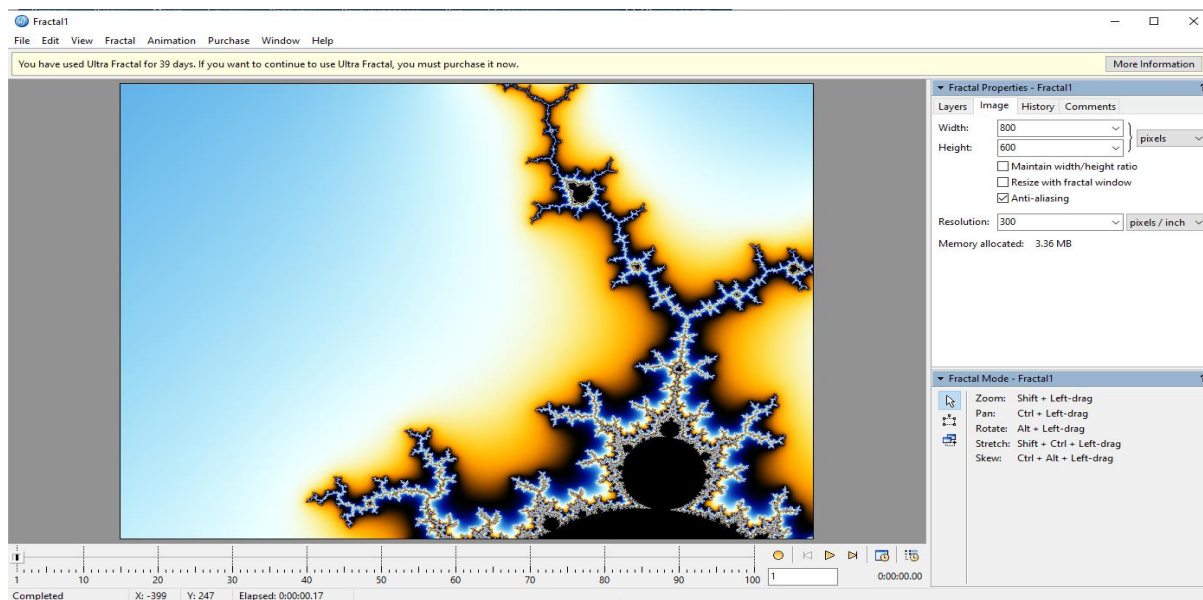


Рис 1.1 Ultra Fractal

Програма Ultra Fractal(Рис 1.1) є одним із лідерів по популярності серед ППФ, ажде вона має зручний користувацький інтерфейс, можливість збереження фракталу в кожен момент зміни та має можливість запису

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				9
Змн.	Арк.	№ докум.	Підпис	Дата		

змін(“анімація”). А також плавний перехід (хоча із погіршення якості) в момент зміни масштабу і можливість задання власної формули. Але незважаючи на зручність інтерфейсу, він має один серйозний недолік – надмірна кількість елементів управління, які можуть заважати роботі. Не зручно реалізована механіка приближення. Також дана програма є комерційною, адже надається доступ до програми на безплатній основі на деякий період.

Основні плюси:

- Зручний користувацький інтерфейс
- Можливість задання власної формули
- Стабільність роботи програми

Основні мінуси:

- Надмірність елементів управління
- Комерційність
- Не зручно реалізація механіки приближення

### **Генератор фракталів від Nadin desing**

Розробник: Надія Трубочкіна

Мова програмування: JScript.

Посилання на ОП: <http://nadin.miem.edu.ru/1111/index.html>

Розглянемо другий аналог Генератор фракталів від Nadin desing:

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				10
Змн.	Арк.	№ докум.	Підпис	Дата		

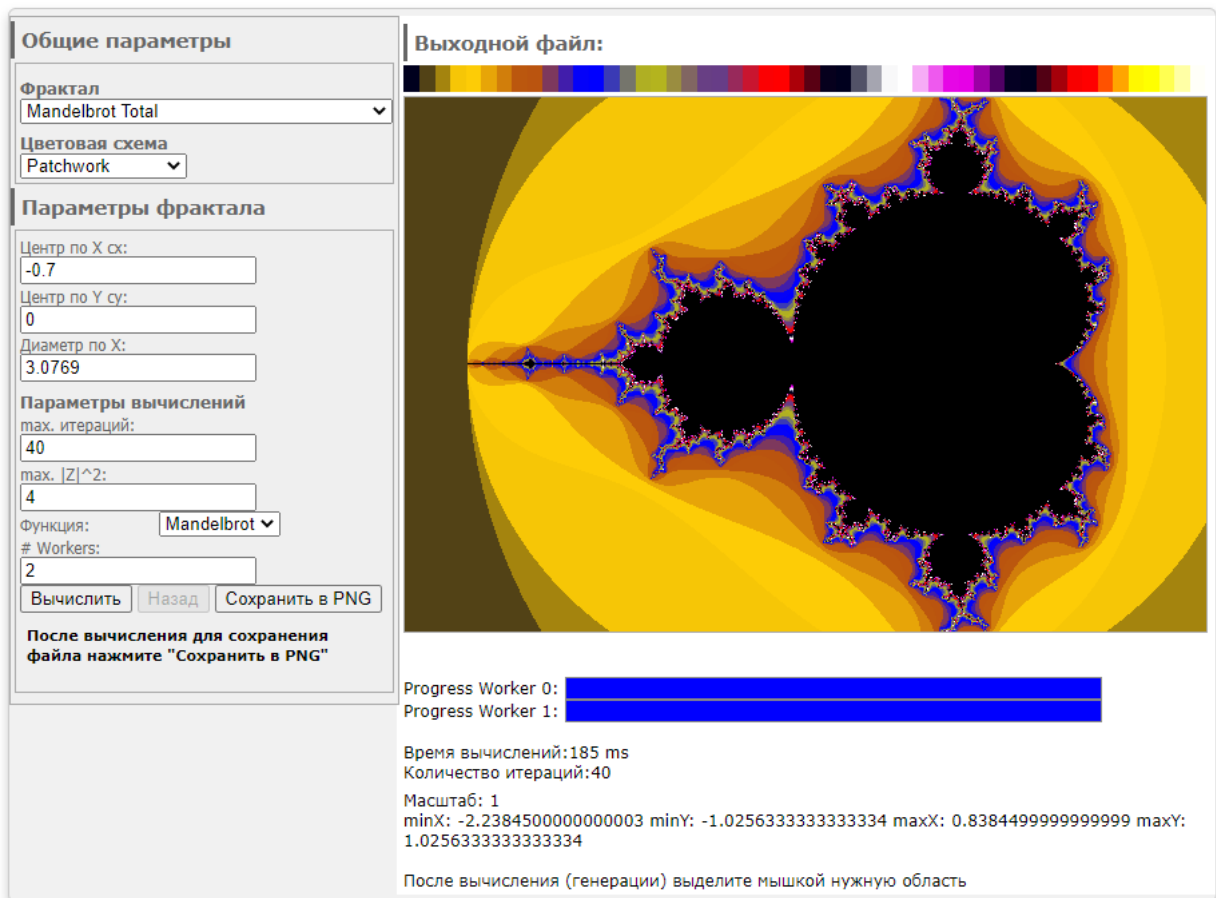


Рис 1.2 Генератор фракталів від Nadin desing.

Програма генератор фракталів від Nadin desing (Рис 1.2) є лідером серед мережових ППФ. Вона має мінімалістичний користувацький інтерфейс, вибір в загальному тільки серед двох фракталів таких, як множина Мандельброта та множина Жульє. Також мінусом є те, що при великому значенні масштабування процес створення відбувається надто довго. Має можливість збереження зображення фракталу тільки в форматі PNG. Механіка масштабування краща ніж у Ultra Fractal.

Основні плюси:

- Зручний користувацький інтерфейс
- Стабільність роботи програми
- Механіка приближення реалізована зручно

		Кормиш Р.І			ДУ «Житомирська політехніка». 21.121.12.000 - ПЗ	Арк.
		Власенко О.В				11
Змн.	Арк.	№ докум.	Підпис	Дата		

Основні мінуси:

- Обмеженість вибору двома фракталами
- Довге виконання обчислень при великому приближенні
- Збереження зображення фракталу тільки в форматі PNG

## Fractal Explorer

Розробник: Sirotinsky Arthur

Посилання: <http://www.fractal-explorer.com/>

Розглянемо третій аналог Fractal Explorer:

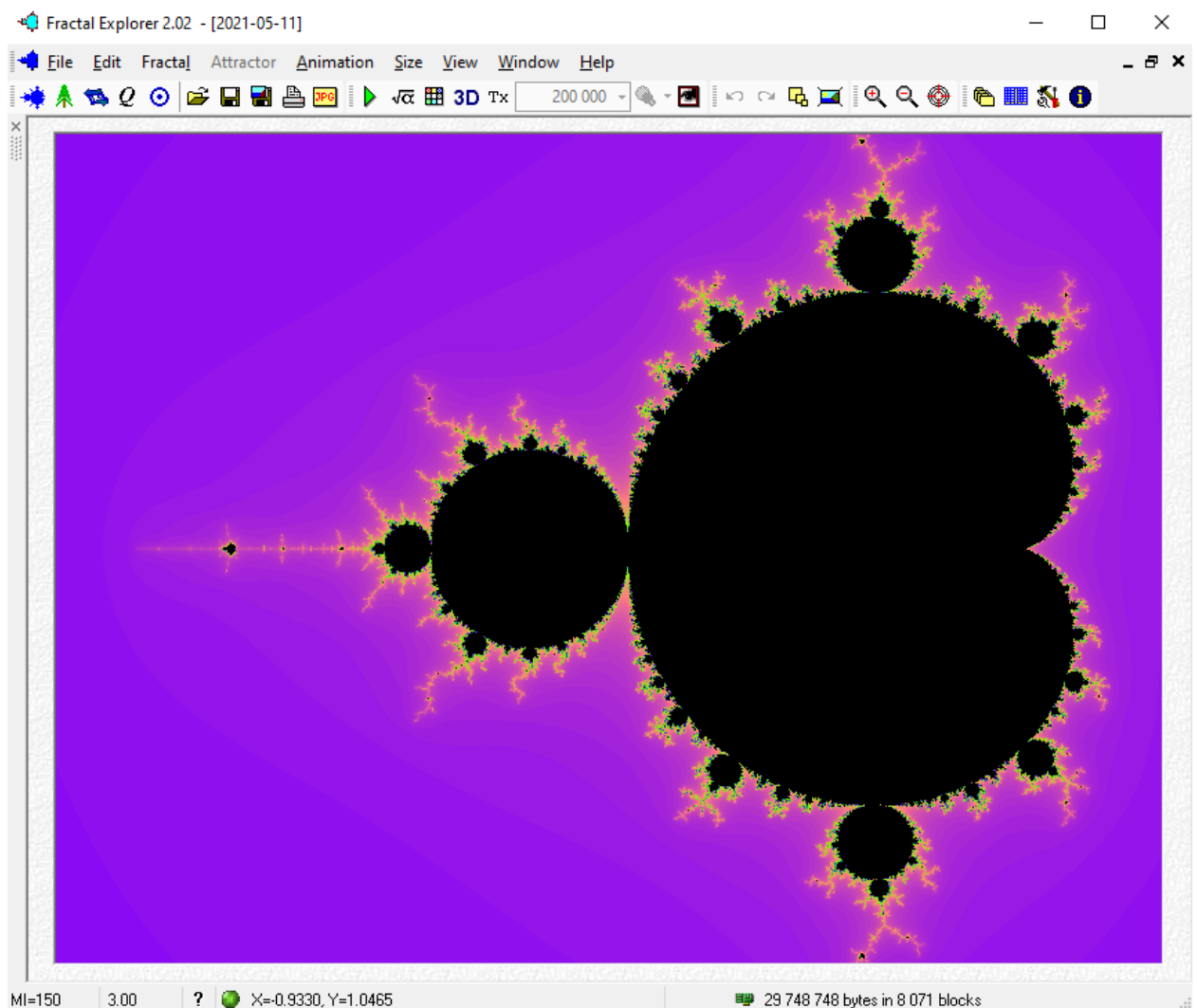


Рис 1.3 Fractal Explorer.

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				12
Змн.	Арк.	№ докум.	Підпис	Дата		

Програма Fractal Explorer (Рис 1.3) займає позиції біля вершини списку по популярності. Вона має широкий вибір фракталів для побудови, можливість роботи з різними фракталами одночасно, присутня анімація та збереження у всіх популярних форматах. Але вона має декілька недоліків таких, як : можливість накладок зображення під час побудови та є наявними елементи комерційності.

Основні плюси:

- Широкий вибір фракталів
- Можливість роботи з різними фракталами одночасно
- Збереження у всіх популярних форматах

Основні мінуси:

- Можливість накладок зображень
- Наявні елементи комерційності

Актуальним напрямком реалізації власного продукту є побудова фракталів методами схожими з методами Ultra Fractal , реалізувати користувацький інтерфейс, уникаючи надмірності Ultra Fractal, але послідувати основним аспектам інтерфейсу цього продукту та звертатися до методів продукту, приведеного на Рис 1.2 та Fractal Explorer, також забезпечити стабільність програми та можливість масштабування .

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				13
Змн.	Арк.	№ докум.	Підпис	Дата		

### Висновки до першого розділу :

У ході виконання першого розділу було поставлено завдання проекту, а саме який має бути функціонал , де буде застосовуватися і визначено актуальність проекту. Також обрано платформу та мову розробки, визначено актуальний напрямок розробки власного продукту. Розглянуто існуючі програмні продукти зі схожою тематикою та більш детально проаналізовано деякі із них, а саме : Ultra Fractal, Генератор фракталів від Nadin desing, та Fractal Explorer.

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				14
Змн.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 2.1 Проектування загального алгоритму роботи програми

На даному етапі розробки розглянемо загальний алгоритм роботи ППФ. Загальна схема роботи ППФ приведена на рисунку 2.1.

Спочатку буде створено функції для створення фракталів насамперед в основні формі, але після створення бібліотек класів буде перероблено функції та перенесено їх до відповідної бібліотеки *FractalClasses*, яка міститиме фрактальні класи : *MBrotSet*, *FractalTree*, *BarnsleyFern* та *CurveDragon*.

Також буде створена друга бібліотека класів, яка містила допоміжні класи *HelperClasser*, яка міститиме клас для збереження кольорів конкретних пікселів *Pixel* та клас для комплексних чисел *Complex*.

Наступним кроком після побудови фракталів буде реалізація їх збереження. Що ми зможемо виконати за допомогою діалогових вікон. А для можливості збереження в різних форматах використано властивість *filter*.

Для можливості та зручності використання ППФ буде додано багато елементів управління та реалізовано “*hot keys*”.

Також для зручності задання кольорів фракталів буде реалізовано окреме вікно та додано можливості генерації градієту і завантаження користувачем відповідного шаблону кольорів.

Тож, ППФ матиме вигляд декількох вікон зв’язаних між собою, які зручний користувацький інтерфейс, де користувач зможе задавати параметри обраних фракталів.

При запуску програми користувач буде бачити перед собою побудовану множину Мандельброта і зможе вибрати інший фрактал в *ComboBox*. Далі вводяться початкові дані і перевіряються на коректність.

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				15
Змн.	Арк.	№ докум.	Підпис	Дата		

Для множини Мандельброта початковими даними є точка, відносно якої будується фрактал, квадрат максимального значення функції та кількості ітерацій. При введенні некоректних даних повинна виводитися іконка з попередженням.

Для фрактального дерева початковими даними є довжина гілки, мінімальна довжина гілки, кількість гілок, градуси відхилю гілок від кореневої гілки, початкові координати та ширина гілки. Кількість гілок в даній ППФ користувач може обирати за допомогою *ComboBox*, який має на вибір кількість гілок від 1 до 4. Далі користувач також може задати задній фон для фракталу.

Для папоротника Барнслі початковими даними є кількість точок, які будуть формувати фрактал, звуження по висоті і ширині, а також задання заднього фону фракталу.

Останнім фракталом на вибір є крива Хартера-Хейтуея, початковими даними для якої є кількість кривих, координати точок початку і кінця цих кривих, ширина кісті, яка задає ширину лінії кривої, кількість ітерацій, яка задає кількість згинів кривої, а також задання заднього фону фракталу.

Після обрання фракталу та введення коректних даних користувач нажиматиме на кнопку побудови, яка розміщена в лівому нижньому кутку вікна.

Під час побудови фракталу під кнопкою буде заповнюватися *progressBar*, який відображатиме процес побудови заданого фракталу. Також є можливість зміни колірної палітри фракталу, за допомогою задання градієнту.

Далі користувач може зберегти створений фрактал та далі продовжити працювати з програмою.

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				16
Змн.	Арк.	№ докум.	Підпис	Дата		



## Загальна схема роботи програми :

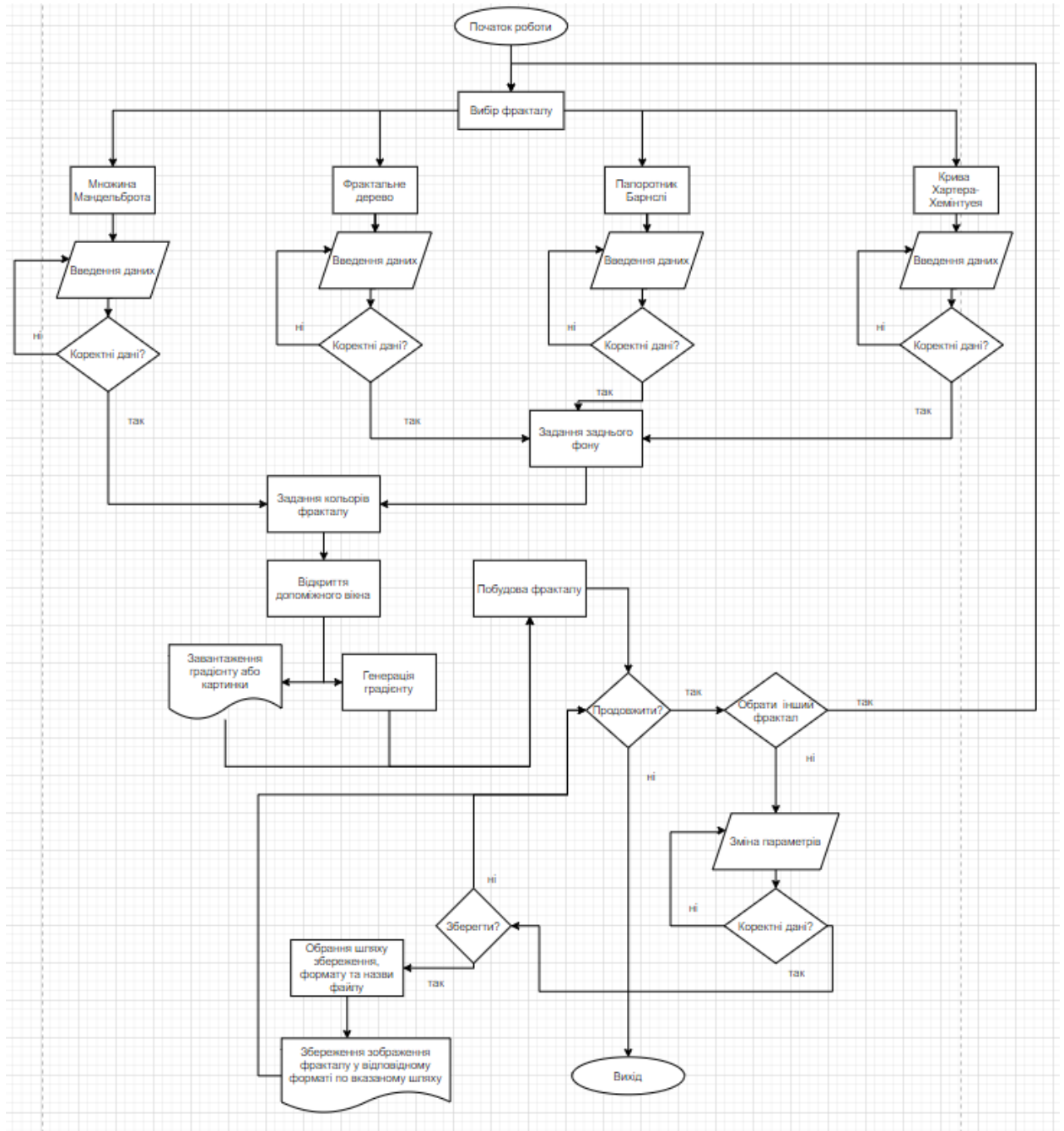


Рисунок 2.1 – Загальна схема роботи програми

Детальні схеми пробудови кожного фракталу наведені нижче.

## Схема множини Мандельброта:

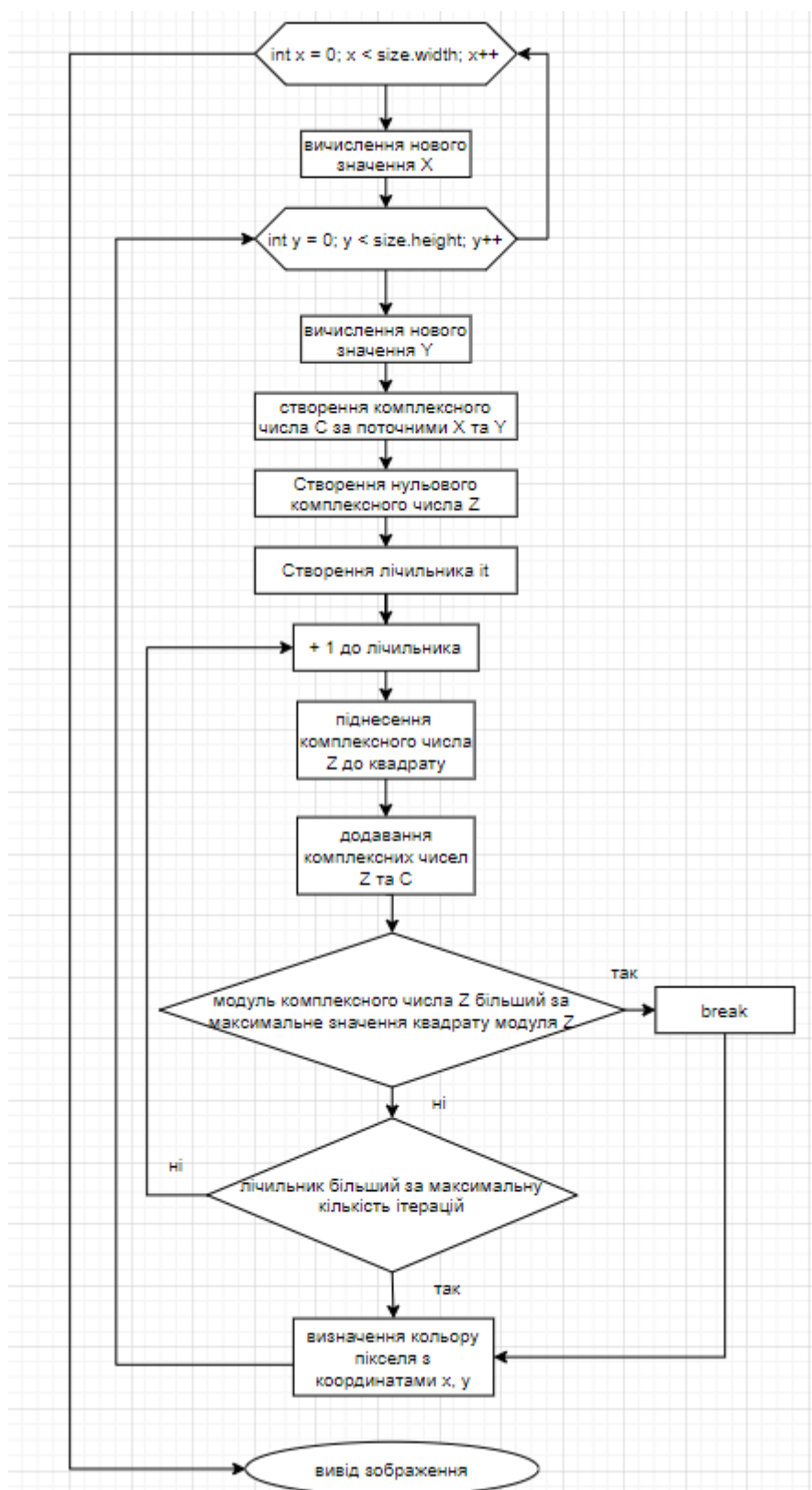


Рисунок 2.2 – Схема множини Мандельброта

Схема фрактального дерева:

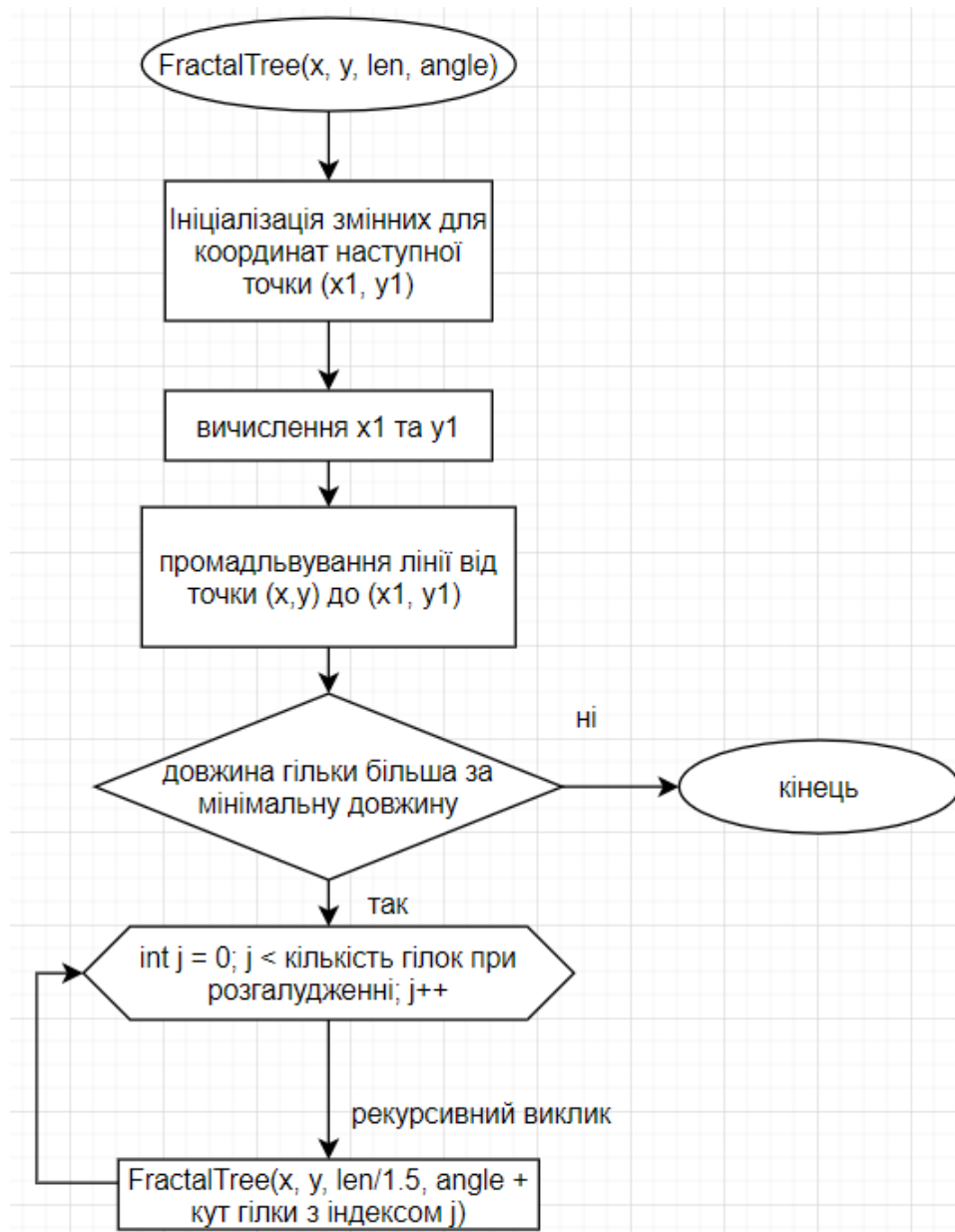


Рисунок 2.3 – Схема фрактального дерева

## Схема папоротника Барнслі:

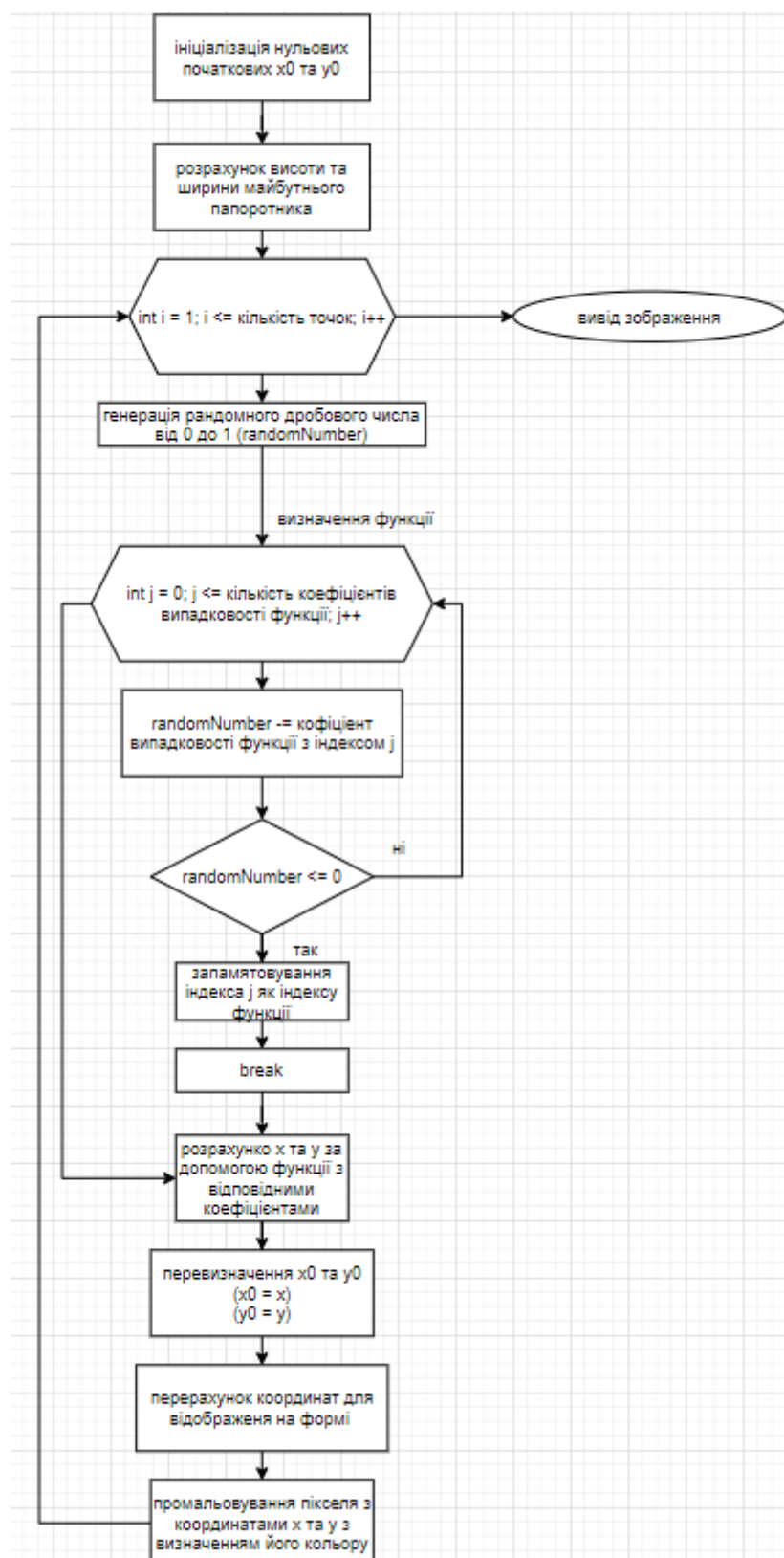


Рисунок 2.4 – Схема папоротника Барнслі

### Схема кривої Хартера – Хейтуея:

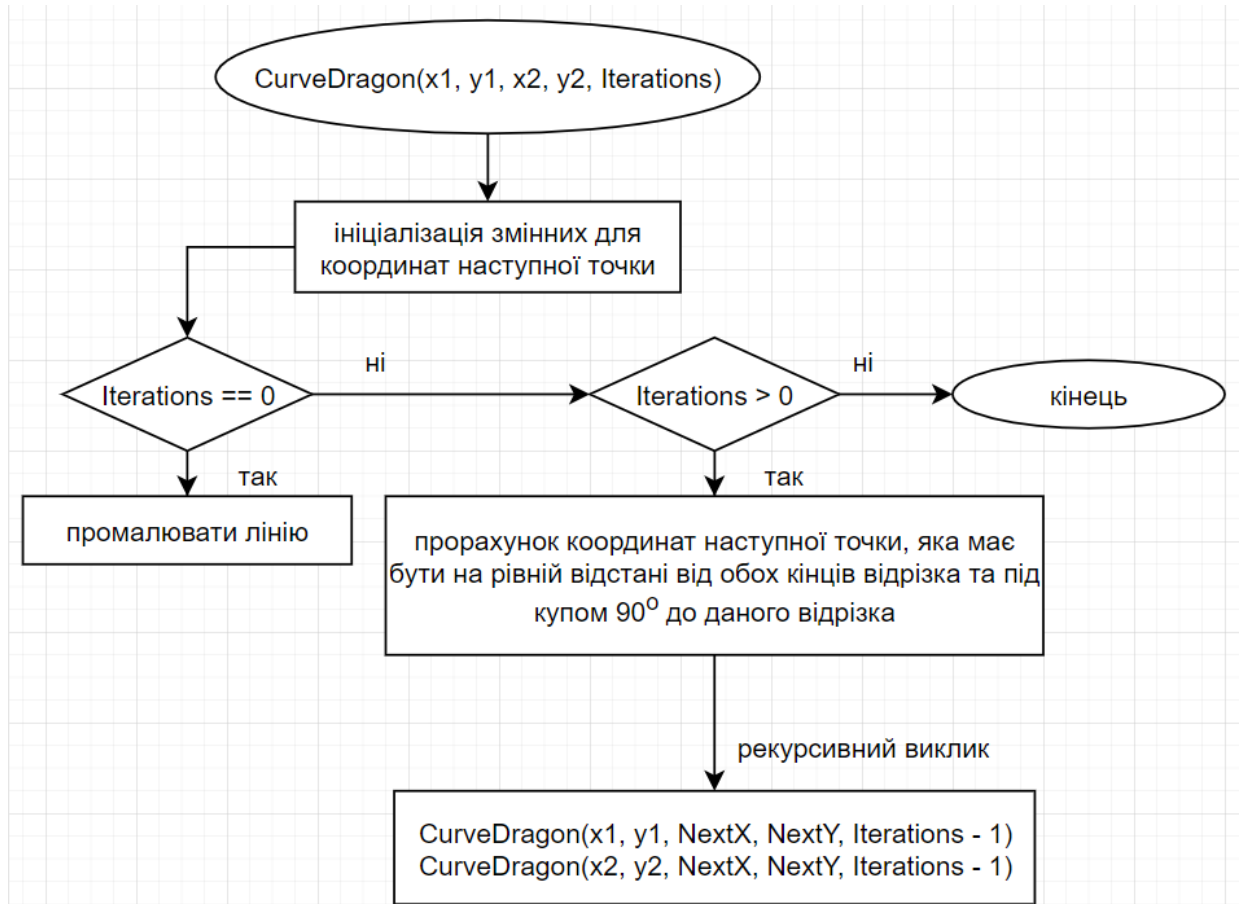


Рисунок 2.5 – Схема кривої Хартера – Хейтуея

## 2.2 Розробка функціональних алгоритмів роботи програми

Залишилося розглянути детальніше алгоритми роботи основних функцій.

Звичайно, основне призначення ППФ – побудова фракталів, тому розглянемо саме це. Для малювання на формі треба використовувати елемент класу Graphics. Для побудови кожного фракталу використовується відповідна йому фрактальна формула, після поітераційного обчислення цієї формули ми отримуємо зображення фракталу. Для зручності огляду будь-якого фракталу потрібно використовувати палітру кольорів(градієнт), що вирізняти різні його елементи.

При обранні у програмі фракталу, задаються початкові значення автоматично, які користувач може змінити. В залежності від того, який обирається фрактал, надаються різні поля вводу даних.

Сам алгоритм полягає в тому, що при нажатті на кнопку побудови фракталу, Створюється об'єкт з відповідним класом, який наслідується від базового класу FractalClasses, виконуються необхідні операції, а після їх виконання передається створене зображення фракталу в *PictureBox*.

Для виконання побудови множини Мандельброта використовуються об'єкти типу *Complex*, які відповідають за комплексні числа, адже формула цього фракталу побудована саме на них. Кольори для відображення кожного фракталу зберігаються в *List* типу *Pixel*.

При роботі зі збереженням та завантаженням даних в ППФ використовувався простий алгоритм перевірки на порожнечу та відповідність параметрам. При збереженні спочатку перевірялося чи зображення фракталу існує і чи існує картинка з заданою користувачем назвою, а потім виконувалося збереження.

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		22

При завантаженні градієнту виконується алгоритм перевірки на порожнечу, а Також для перевірки на відповідність параметрам було додано алгоритм перевірки розрішення картинки. За умовою якого ширина картинки має становити не менше за 100 пікселів. Якщо умова не виконується відображається відповідне попередження.

Для побудови фракталів було розроблено алгоритми генерації відповідних фракталів за допомогою опису відповідних фрактальних формул. При зміні параметрів запускаються відповідні алгоритми які залежать як від даних введених користувачем так і від розміру віконного додатку. Також передбачені алгоритми для уникнення виникнення накладок як програмного коду так і візуального відображення, які забезпечують зручність використання продукту та його стабільність.

Діаграма класів та компонентів :

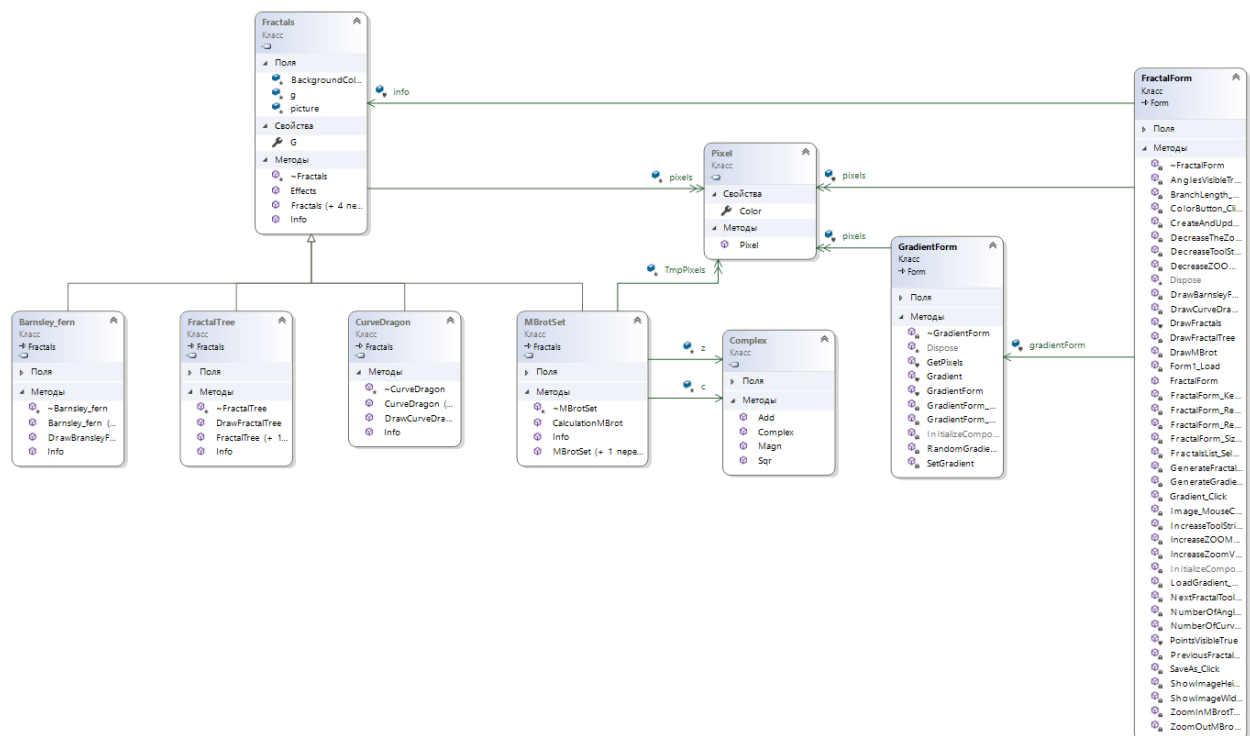


Рисунок 2.6 – Діаграма класів та компонентів

## 2.3 Розробка програмного забезпечення

Структура рішення проекту :

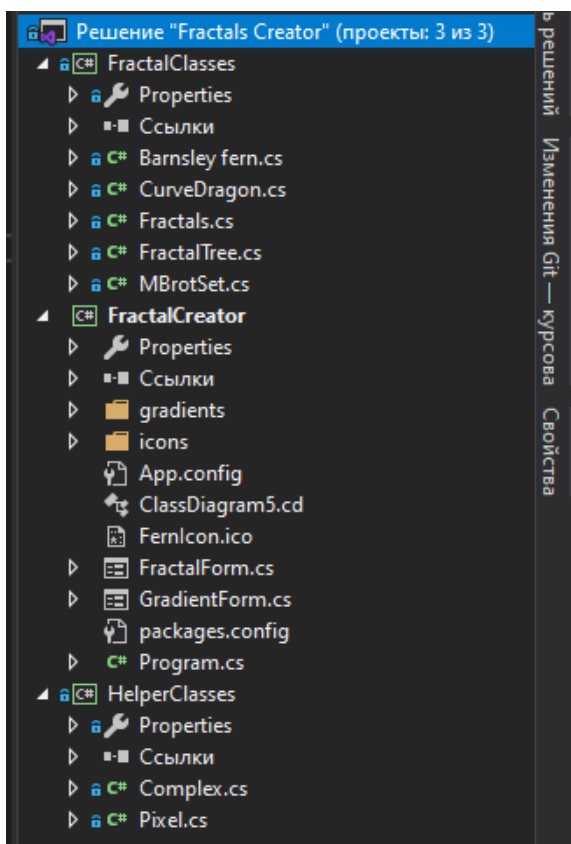


Рисунок 2.7 – Структура рішення проекту

*FractalClasses* – це бібліотека класів, яка містить , як не дивно, фрактальні класи. В ній містяться 5 класів: *BurnsleyFern*, *CurveDragon*, *FractalTree*, *MBrotSet* і базовий клас *Fractals*.

*HelperClasses* - це бібліотека класів, яка містить, допоміжні класи. В ній містяться 2 класи: *Complex* та *Pixel*.

Фрактальні класи і є основними в даному проекті. Тож розглянемо їх детальніше.

Клас *MBrotSet* відповідає за побудову множини Мандельброта, яка описується за допомогою відповідної формули :  $f_c(z) = z^2 + c$ , де  $c$  і  $z$  комплексні числа. Для можливості виконання дій над комплексними числами було створено клас *Complex*.

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				24
Змн.	Арк.	№ докум.	Підпис	Дата		



Цей клас має такі методи, як *Sqr()*, який забезпечує піднесення комплексного числа до квадрату,

```
public void Sqr()
{
    double tmp = (Re * Re) - (Im * Im);
    Im = 2.0d * Re * Im;
    Re = tmp;
}
```

*Magn(Magnitude)()*, який повертає модуль комплексного числа,

```
public double Magn()
{
    return Math.Sqrt((Re * Re) + (Im * Im));
}
```

і метод *Add()*, який додає два комплексних числа.

```
public void Add(Complex c)
{
    Re += c.Re;
    Im += c.Im;
}
```

Саме ці методи забезпечують можливість виконання операцій над комплексними числами. Звичайно, тут описані не всі методи виконання дій над комплексними числами, але в даному проєкті їх достатньо.

Якщо при роботі програми обрана множина Мандельброта, тоді викликається метод *CalculationMBrot()*, який повертає об'єкт типу *Bitmap* з зображенням фракталу. Де основною частиною є розрахунок комплексного числа  $z$ :

```
do
{
    it++;
    z.Sqr();
    z.Add(c);
    if (z.Magn() > maxZ)
    {
        break;
    }
} while (it < UserIt);
```

Саме цей цикл і є реалізацією фрактальної формули Мандельброта.

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				25
Змн.	Арк.	№ докум.	Підпис	Дата		

Також поверає *Bitmap* метод *DrawBransleyFern()*, який буде папоротник Барнслі. Основним циклом якого є вирахування можливостей формул заданих коефіцієнтами:

```
double randomNum = random.NextDouble();
for (int j = 0; j < probability.Length; j++)
{
    randomNum -= probability[j];
    if (randomNum <= 0)
    {
        FunctionIndex = j;
        break;
    }
}
```

, де коефіцієнти можливостей задані масивом `float[] probability = new float[4] { 0.01f, 0.06f, 0.08f, 0.85f }`; а формули задані коефіцієнтами формул

```
float[,] Coefficient = new float[4, 6]
{
    {0, 0, 0, 0.16f, 0, 0},
    {-0.15f, 0.28f, 0.26f, 0.24f, 0, 0.44f},
    {0.2f, -0.26f, 0.23f, 0.22f, 0, 1.6f},
    {0.85f, 0.04f, -0.04f, 0.85f, 0, 1.6f}
};
```

, де кожна формула задана 6 коефіцієнтами, які задовільняють формулу

$$f(x, y) = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}$$
, де кожна літера відповідає індексації елементу в масиві, а початковим  $x$  і  $y$  беруться точки початку побудови.

Інші методи, а саме *DrawCurveDragon* та *DrawFractalTree* не повертають такого об'єкту, адже виконуються за допомогою рекурсивних функцій та мають ти *void*.

Основною частиною методу *DrawFractalTree* є розрахунок наступної точки для побудови лінії:

```
int x1, y1;
x1 = (int)(x + len * Math.Sin((2 * Math.PI * angle) / 360.0));
y1 = (int)(y + len * Math.Cos((2 * Math.PI * angle) / 360.0));
```

, де *angle* це кут нахилу гілки відносно попередньої гілки, ця дія виконується циклічно зі зміною довжини(*len*) та кута(*angle*) лінії(гілки) , що і забезпечує побудову фракталу в формі дерева.

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				26
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        for (int j = 0; j < angles.Length; j++)
        {
            DrawFractalTree(x1, y1, (int)(len / 1.5), angle + angles[j],
progress);
        }

```

Зазвичай будують фрактальне дерево з двома основними вітками, але в даному проекті є можливість спробувати побудувати як із однієї так із трьох чи чотирьох гілок. Швидкість виконання побудови залежить від кількості новостворених гілок, чим більша різниця між довжинами гілок та чим їх більше тим повільніше відбувається процес побудови. Для побудови фрактала Хартера-Хейтуея використовується не складний алгоритм, який полягає в діленні відрізка навпіл під кутом  $90^\circ$

```

NextX = (int)((x1 + x2) / 2 + (y2 - y1) / 2);
NextY = (int)((y1 + y2) / 2 - (x2 - x1) / 2);

```

, де *NextX* і *NextY* - це координати точки поділу лінії, в яких формулах яких перша частина це точка середини відрізка, а друга координати точки відносно лінії під кутом  $90^\circ$ . Ці операції виконуються циклічно, що і забезпечує самоподібність і оптичну складність даного фракталу.

Крім фрактальних методів, є також важливий метод отримання кольору пікселів з градієнту чи з картинки:

```

public List<Pixel> GetPixels(Bitmap bitmap)
{
    List<Pixel> pixels = new List<Pixel>(bitmap.Width);
    for (int x = 0; x < bitmap.Width; x++)
    {
        pixels.Add(new Pixel()
        {
            Color = bitmap.GetPixel(x, bitmap.Height / 2) // отримання кольору
ряду пікселів з градієнту
        });
    }
    return pixels;
}

```

Цей метод, можна так сказати, вирізає один ряд пікселів посередині картини чи градієнту і передає його у вигляді списку кольорів. Потім цей список використовується як градієнт або набір кольорів для розфарбування фракталів.

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				27
Змн.	Арк.	№ докум.	Підпис	Дата		

Крім можливості задання розфарбування фракталу за допомогою зовнішніх ресурсів , є також можливість генерації градієнту для розфарбування фракталу, насамперед, в програмі. Це забезпечує метод SetGradient():

```
private void SetGradient(Bitmap Gradient, int r, int g, int b, int NoChangeIndex)
{
    int[] change = new int[3];
    change[NoChangeIndex] = 0;
    for (int x = 0; x < Gradient.Width; x++)
    {
        for(int i = 0; i < change.Length; i++)
        {
            if(i != NoChangeIndex)
            {
                change[i] = x;
            }
        }
        for (int y = 0; y < Gradient.Height; y++)
        {
            Gradient.SetPixel(x, y, Color.FromArgb(((r + change[0]) / 3) % 255,
            ((g + change[1]) / 3) % 255, ((b + change[2]) / 3) % 255));
        }
    }
}
```

, якому в якості параметрів 3 змінних відповідних основних кольорів в системі RGB і індекс параметра який залишається статичним. Це забезпечує можливість створення кольорових градієнтів. В цьому методі в першому внутрішньому циклі визначається саме яка змінна не буде змінюватись. А в другому саме попіксельне створення градієнту.

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				28
Змн.	Арк.	№ докум.	Підпис	Дата		

### Висновки до 2 розділу :

Спроековано загальну схему програми, діаграму класів. Визначено та детально описано основні можливості програми. Пояснено алгоритми найважливіших методів ППФ та приведено код їх основних частин. Приведено приклади коду деяких методів. Роз'яснено деякі нюанси роботи програми. Показано структуру проекту.

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				29
Змн.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 3 ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ

### 3.1 Опис роботи з програмним додатком (Опис інтерфейсу)

Після запуску програми з'являється головне вікно (Рис 3.1), де починається побудова множини Мандельброта з параметрами по замовчуванню. Більше частину вікна займає PictureBox, де і відмальовується фрактал. Зліва від нього розташоване меню, де задаються параметри фракталів.

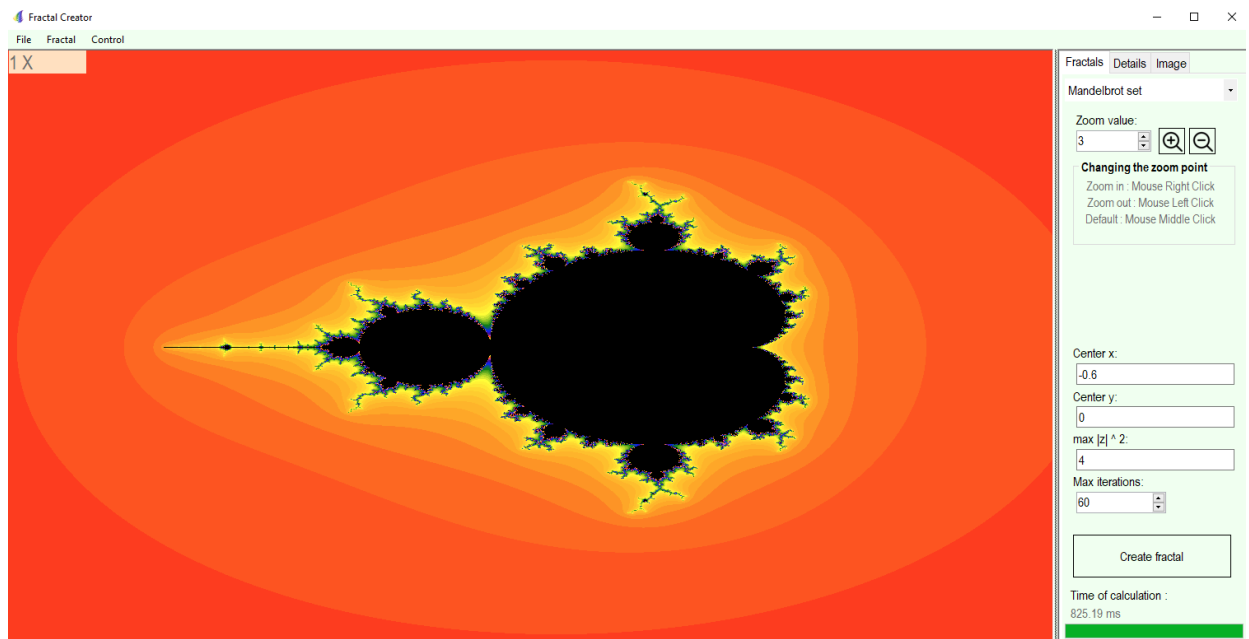


Рис 3.1 Головне вікно

Меню має 3 вкладки: Fractals (вкладка з параметрами фракталів), Details (вкладка з інформацією про обраний фрактал) та вкладка Image, де виводяться розміри створеного зображення.

У вкладці Fractals розміщений ComboBox, де користувач може обирати вид фракталу (по замовчуванню це Mandelbrot set (множина Мандельброта)).

		Кормиш Р.І			ДУ «Житомирська політехніка». 21.121.12.000 - ПЗ	Арк.
		Власенко О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		30

При обранні множини Мандельброта користувач може змінити дані по замовчуванню, а саме: координати множини відносно лівого краю вікна, максимальний модуль квадрата числа  $z$  та кількість ітерацій відповідної формули. Потім користувач має натиснути на кнопку нижче, де написано Create fractal, яка відповідає за запуск процесу побудови. Підчас побудови користувач може відслідкувати процес за допомогою progressBar, який розміщений нижче кнопки. Після завершення побудови з'явиться зображення фракталу та нижче кнопки з'явиться час побудови даного фракталу.

Коли множина Мандельброта побудована користувач може досліджувати фрактал за допомогою миші(Рис 3.2), кнопок меню(Рис 3.3) та hotkeys.

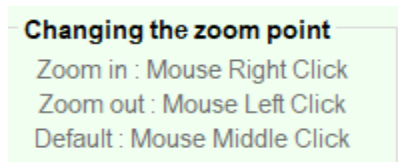


Рис 3.2 Керування за допомогою миші

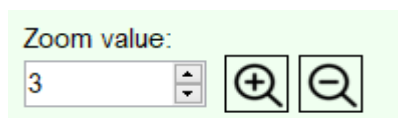


Рис 3.3 Кнопки меню для дослідження фракталу

При нажатті на кнопки в меню, зображення приближується на відповідне значення поля Zoom value, а точкою приближення є центр зображення. При керуванні мишкою точка кліку задає центр і відбувається той же процес, що й при нажатті кнопок меню, але зі зміною центра. Нажаття лівої кнопки миші відповідає за приближення, а правої за віддалення. Такожт при нажатті на колесико миші, фрактал повертається до початкового стану.

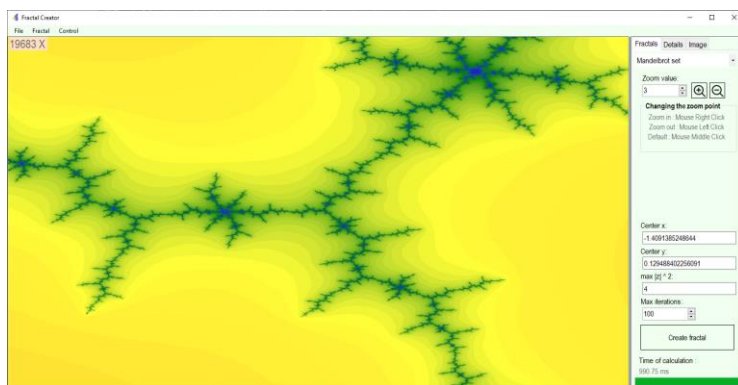


Рис 3.4 Множина Мандельброта з приближенням в 243 рази та зміненими параметрами

При обранні в ComboBoxі фракталу під назвою Fractal tree(Рис 3.5), користувач бачить меню з іншими полями, які також мають параметри по замовчуванню.

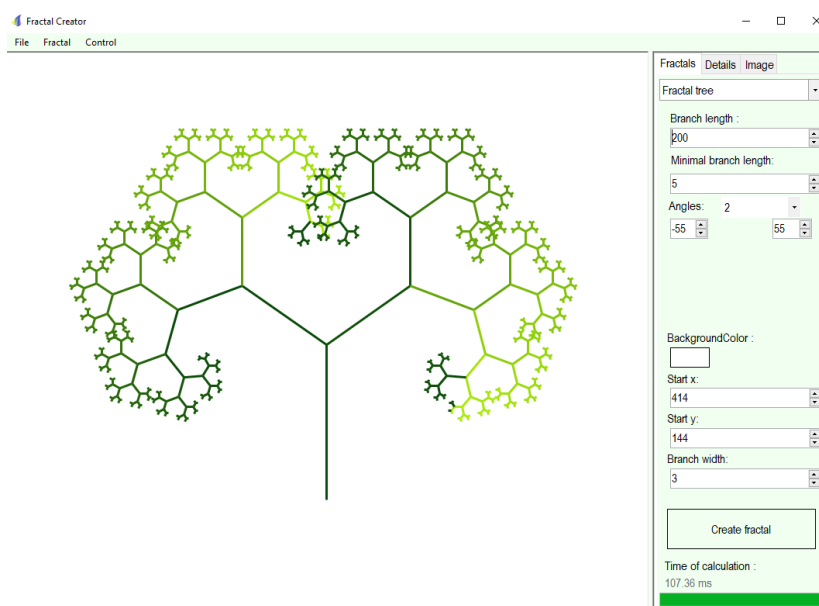


Рис 3.5 Фрактальне дерево

Перші два поля відповідаю за довжину гілки – перше поле за початкову довжину, друге за мінімальну довжину. Нижче розміщений ComboBox, де користувач обирає кількість гілок. Потім користувач вписує значення кутів цих гілок. При позитивних значеннях гілки відхиляються в правий бік, а при негативних - в лівий. Далі користувач вводить значення початкової точки у відповідні поля і може задати товщину вітки. Далі все як і при побудові в попереднього фракталу.

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				32
Змн.	Арк.	№ докум.	Підпис	Дата		



При обранні користувачем в ComboBoxі фракталу під назвою Fern Barnsley (Рис 3.6), користувач бачить побудований папоротник із 50000 ТОЧОК.



Рис 3.6 Папоротник Барнслі

Для зміни кількості точок користувач має змінити число в полі Number of points. Також він може задати звуження по обом осям, записавши значення в поля Horizontal constriction и Vertical constriction, які відповідають за горизонтальне і вертикальне звуження відповідно (Рис 3.7).

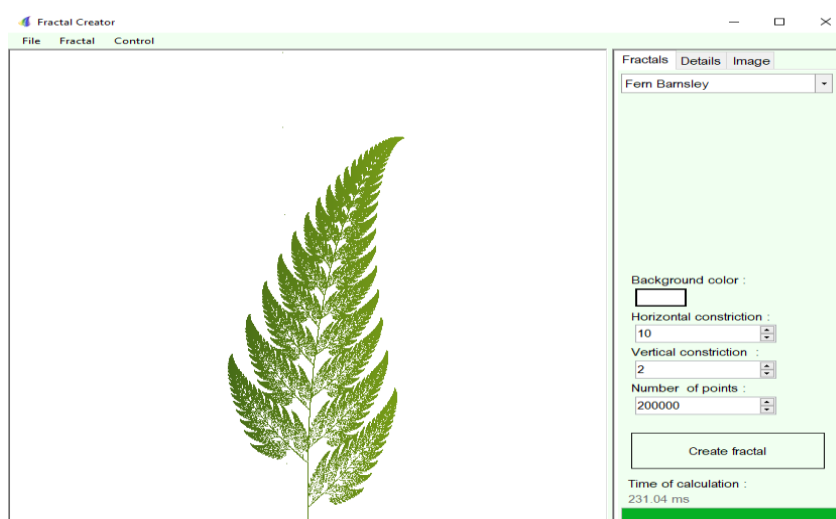


Рис 3.7 Папоротник Барнслі зі зміненими параметрами

При обранні користувачем в ComboBoxі фракталу під назвою Curve of Dragon (Рис 3.8), бачить декілька побудованих кривих Хартера-Хейтуея.

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		33

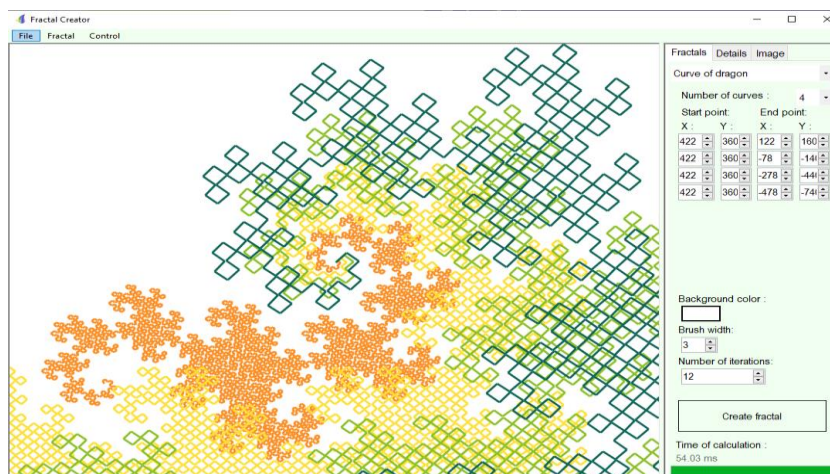


Рис 3.8 Фрактал Хартера-Хейтуея (Крива дракона)

В меню користувач може змінити параметри кривих, такі як : кількість кривих, точки початку і кінця , ширину ліній та кількість ітерацій, змінюючи значення відповідних полів.

Також в трьох попередніх фракталах користувач може задавати задній фор фракталу, якщо він потрібен. Додаткове управління ППФ та фракталів користувач може подивитися на панелі інструментів у вкладках Control та Fractal. Також там він може знайти всі комбінації клавіш. Якщо користувач знайшов зображення фракталу, яке йому потрібне, він може зберегти його або з панелі інструментів або комбінацією клавіш *Ctrl + Shift + S*.

Для зміни колірної палітри фракталу користувач може відкрити допоміжне вікно(Рис 3.9) на вкладці *Fractal > Open gradient window* або комбінацією клавіш *Ctrl + Shift + X*.

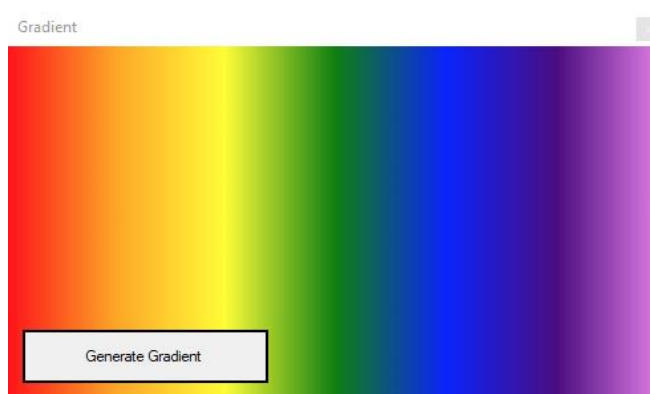


Рис 3.9 Вікно для градієнту

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				34
Змн.	Арк.	№ докум.	Підпис	Дата		

В даному вікні при нажатті на кнопку Generate Gradient або комбінації клавіш Ctrl + Z, створюється градієнт. Також коли користувач завантажує зовнішній ресурс, він відображається в даному вікні(Рис 3.10)



Рис 3.10 Вікно для градієнту з завантаженою картинкою

Саме із заданого в даному вікні зображення зчитується колірна палітра для фракталів.

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				35
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3.2 Тестування роботи програмного забезпечення

Програмний додаток передбачає помилки та попередження при введенні користувачем, для зручного користування програмою. Наприклад, при введенні некоректних даних, при обраному фракталі множина Мандельброта, в поля з координатами множини (Рис 3.11) (Рис 3.12) та в поле з максимальними квадратом модуля числа (Рис 3.13) виникають помилки з відповідними повідомленнями.

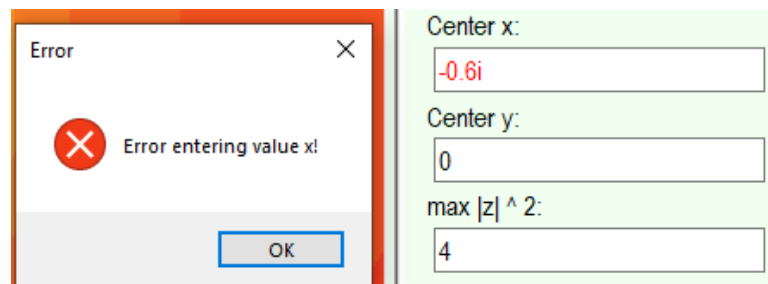


Рис 3.11 Помилка при введенні значення x

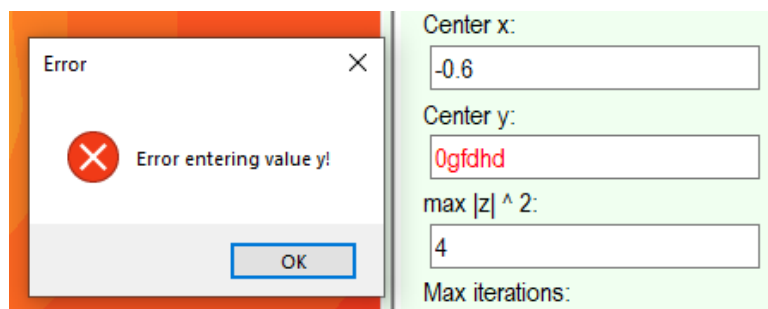


Рис 3.12 Помилка при введенні значення y

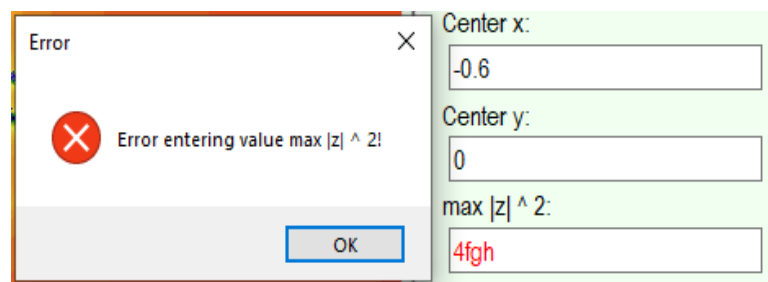


Рис 3.13 Помилка при введенні значення максимального квадрата модуля числа z

При завантаженні градієнту або зображення та при невідповідності розміру зображення, з'являється попередження (Рис 3.14).

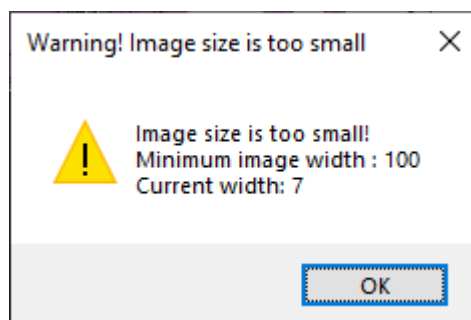


Рис 3.14 Попередження про невідповідність розміру зображення

А при успішному завантаженні та збереженні з'являються відповідні повідомлення (Рис 3.15).

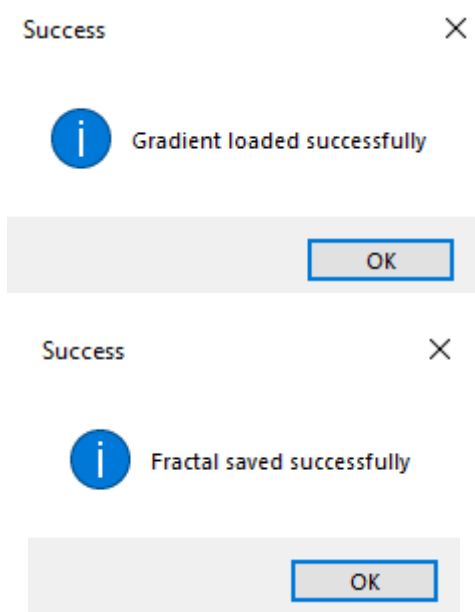


Рис 3.15 Повідомлення про збереження та завантаження

Також під час побудови фракталу деякі елементи інтерфейсу, функції та методи відключаються для уникнення накладок зображення та запобігання виникнення помилок.

### Висновки до 3 розділу :

Було детально описано роботу програмного додатку, користувацький інтерфейс, очікувані дії користувача та їх результати. Також було розписано моменти уникнення помилок завдяки допоміжним перевіркам при введенні даних або при роботі з зображеннями. Завдяки цим перевіркам вдалося уникнути частих та серйозних помилок. Наведено приклади обробки помилок.

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				38
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

Під час написання даного курсового проекту було отримано навички роботи із середою розробки Windows Forms. А саме було розглянуто роботу з графічними елементами.

В першому розділі курсового проекту, було проаналізовано знайдені ППФ зі схожим функціоналом та інтерфейсом, та було сформовано уявлення про ППФ, що розроблюється. Було виявлено основні потреби для написання даного програмного додатку.

В другому розділі курсового проекту, було проведено розробку функціональних і загальних частин ППФ. При розробці функціонального алгоритму, розробили уявлення про методи розробки програмного коду додатку. На заключному етапі, розроблено програмний код спроектованого додатку. В результаті, отримано готовий програмний продукт, який виконує всі заплановані функції і задачі.

В третьому розділі курсового проекту було описано методи користування створеної ППФ, були пояснені і продемонстровані всі створені елементи інтерфейсу в формі. Також було проведено тестування, тобто ППФ було запущено в реальних умовах її користування. В ході тестування проблем не виникало.

В результаті виконання курсового проекту отримано ППФ, яка повністю відповідає запланованому функціоналу, умовам та вимогам, поставленим на початку проектування. ППФ є зрозумілою та зручною в користуванні, адаптивною та швидкодіючою.

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				39
Змн.	Арк.	№ докум.	Підпис	Дата		

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Об'єктно-орієнтоване програмування [Електронний ресурс] / Освітній портал ДУ «Житомирська політехніка» – 2020. – Режим доступу до ресурсу: <https://learn.ztu.edu.ua/course/view.php?id=1628>
2. Технології програмування. Мова С# : навч. посібн/ В.В. Томашевський. – Житомир: ЖВІ НАУ, 2012. – 484 с.
3. Шилдт Герберт С# 4.0: полное руководство.: Пер. с англ. — М.: ООО "И.Д. Вильямс", 2011. — 1056 с.: ил.
4. Эндрю Т. Язык Программирования С# 5.0 и платформа .NET 4.5 / Троелсен Эндрю., 2013. – 1168 с. – (6-е издание).
5. Павловская Т.А. Программирование на языке высокого уровня / Т.А. Павловская – СПб.: Питер, 2011, – 461с.: ил.
6. Джеймс Глейк Хаос. Создание новой науки / Джеймс Глейк, 1987. – 416 ст.
7. Види фракталів та методи їх створення. [Електронний ресурс] – Режим доступу: <https://sites.google.com/site/fraktali22374/home/vidi-fraktaliv-ta-metodi-ieh-stvorennja>
8. Асинхронное программирование . [Електронний ресурс] – Режим доступу: <https://metanit.com/sharp/tutorial/13.3.php>
9. Введение во фракталы [Електронний ресурс] – Режим доступу: <http://algotlist.ru/graphics/fracart.php>
10. OpenFileDialog и SaveFileDialog [Електронний ресурс] – Режим доступу: <https://metanit.com/sharp/windowsforms/4.20.php>

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				40
Змн.	Арк.	№ докум.	Підпис	Дата		



# ДОДАТКИ

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		41

## Програмний код:

## Бібліотека класів FractalClasses :

Class Fractals.cs :

```

using System;
using System.Collections.Generic;
using HelperClasses;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Windows.Forms;

namespace FractalClasses
{
    public class Fractals
    {
        protected List<Pixel> pixels;
        protected Bitmap picture;
        protected Graphics g;
        protected Color BackgroundColor;
        public Graphics G
        {
            set
            {
                if (picture == null)
                {
                    g = null;
                }
                else
                {
                    g = value;
                }
            }
        }
        public Fractals()
        {
            this.picture = null;
            this.pixels = null;
        }
        public Fractals(Bitmap picture, List<Pixel> pixels, Color BackgroundColor) :
        this(picture, pixels)
        {
            this.BackgroundColor = BackgroundColor;
        }
        public Fractals(Bitmap picture, List<Pixel> pixels):this(picture)
        {
            this.pixels = pixels;
        }
        public Fractals(Bitmap picture):this()
        {
            this.picture = picture;
            G = Graphics.FromImage(picture);
        }
        public Fractals(Bitmap picture, Color BackgroundColor) : this(picture)
        {
            this.BackgroundColor = BackgroundColor;
        }
    }
}

```

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				42
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        public void Effects(Graphics g, Color BackgroundColor)
        {
            g.Clear(BackgroundColor);
            g.SmoothingMode = SmoothingMode.AntiAlias;
        }
        public virtual void Info(TextBox info) { }
        ~Fractals()
        {
            Console.WriteLine("Class Fractals is cleared");
        }
    }
}

```

### Class MBrotSet.cs :

```

using System;
using System.Collections.Generic;
using System.Drawing;
using System.Windows.Forms;
using HelperClasses;

namespace FractalClasses
{
    public class MBrotSet:Fractals
    {
        protected List<Pixel> TmpPixels;
        protected Size size;
        protected PictureBox gradientBox;
        protected int Iterations;
        protected Complex c, z;

        public MBrotSet(Bitmap picture, List<Pixel> pixels, Size size, PictureBox
gradientBox,int Iterations):base(picture, pixels)
        {
            this.size = size;
            this.Iterations = Iterations;
            this.gradientBox = gradientBox;
            TmpPixels = pixels;
        }

        public MBrotSet() : base(){}

        public Bitmap CalculationMBrot(double hx, double hy, double x_, double y_, double
maxZ, double SizeArea, int CountColors, ProgressBar progress)
        {
            progress.Invoke(new Action(() => // делегат для відображення progressBar
            {
                progress.Maximum = size.Width;
            }));

            int UserIt = Iterations;
            int change;
            int[] ColorIndex = new int[CountColors];
            int i = 0;
            for (int p = 0; p < gradientBox.Image.Width; p++)
            {
                if (p % (int)(gradientBox.Image.Width / ColorIndex.Length) == 0)
                {
                    if (i >= ColorIndex.Length)
                    {
                        break;
                    }
                }
            }
        }
    }
}

```

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				43
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        ColorIndex[i] = p;
        i++;
    }
}
if (pixels.Count > gradientBox.Width)
{
    for (int p = 0; p < gradientBox.Width; p++)
    {
        pixels[p].Color = TmpPixels[(int)(p * (pixels.Count /
gradientBox.Width))].Color;
    }
}

for (int x = 0; x < size.Width; x++)
{
    x_ = (hx - SizeArea / 2) + x * (SizeArea / size.Width);
    for (int y = 0; y < size.Height; y++)
    {
        y_ = (hy - SizeArea / 2) + y * (SizeArea / size.Height);

        c = new Complex(x_, y_);
        z = new Complex(0, 0);

        int it = 0;

        do
        {
            it++;
            z.Sqr();
            z.Add(c);
            if (z.Magn() > maxZ)
            {
                break;
            }
        } while (it < UserIt);

        // coloring the set
        if (it < UserIt)
        {
            change = it % ColorIndex.Length;
            picture.SetPixel(x, y, pixels[ColorIndex[change]].Color);
        }
        else
        {
            picture.SetPixel(x, y, Color.FromArgb(0, 0, 0));
        }
    }
    progress.Invoke(new Action(() => // делегат для зміни progressBar
    {
        progress.PerformStep();
    }));
}
return picture;
}

public override void Info(TextBox info)
{
    info.Text = "The Mandelbrot set is the set of complex numbers C" +
        " for which the function f(z)=z^2 + c does not diverge when iterated" +
        " from z = 0, i.e., for which the sequence f(0), f(f(0)) etc.," +
        " remains bounded in absolute value. Its definition is credited" +
        " to Adrien Douady who named it in tribute to the mathematician " +

```

		Кормиш Р.І			ДУ «Житомирська політехніка». 21.121.12.000 - ПЗ	Арк.
		Власенко О.В				44
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        "Benoit Mandelbrot, a pioneer of fractal geometry." +
        Environment.NewLine +
        Environment.NewLine +
        "Visually, inside the Mandelbrot set," +
        " an infinite number of elementary figures can be distinguished," +
        " the largest of which is in the center - the cardioid." +
        " There is also a set of ovals related to the cardioid," +
        " the size of which gradually decreases, tending to zero." +
        " Each of these ovals has its own set of smaller ovals," +
        " the diameter of which also tends to zero, etc." +
        Environment.NewLine +
        Environment.NewLine +
        " This process continues indefinitely, forming a fractal." +
        " It is also important that these processes of figure branching" +
        " do not completely exhaust the Mandelbrot set: if we consider" +
        " additional "branchings" with magnification, then in them" +
        " you can see your cardioids and circles that are not associated" +
        " with the main figure.";
    }

    ~MBrotSet()
    {
        Console.WriteLine("Class MBrotSet is cleared");
    }
}

```

#### Class FractalTree.cs :

```

using System;
using System.Collections.Generic;
using System.Drawing;
using System.Windows.Forms;
using HelperClasses;

namespace FractalClasses
{
    public class FractalTree : Fractals
    {
        protected double[] angles;
        protected int minLen;
        protected int BranchWidth;
        private int i = 0;

        public FractalTree(Bitmap picture, List<Pixel> pixels, double[] angles, int
minLen, int BranchWidth, Color BackgroundColor):base(picture, pixels, BackgroundColor)
        {
            this.angles = angles;
            this.minLen = minLen;
            this.BranchWidth = BranchWidth;
        }

        public FractalTree() : base()
        {
        }

        public void DrawFractalTree(int x, int y, int len, double angle, ProgressBar
progress)
        {
            if (i == 0)

```

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				45
Змн.	Арк.	№ докум.	Підпис	Дата		

```

{
    base.Effects(g, BackgroundColor);
    progress.Invoke(new Action(() => // делегат для відображення progressBar
    {
        progress.Maximum = 0;
        double count = 0;
        double tmpCount = len;
        while (tmpCount > minLen)
        {
            tmpCount /= 1.5;
            count++;
        }
        if (angles.Length == 2)
        {
            for (int i = 0; i < count; i++)
            {
                progress.Maximum += (int)(Math.Pow(2, i) /
Math.Pow(angles.Length, 2));
            }
        }
        else if(angles.Length == 3)
        {
            for (int i = 0; i < count; i++)
            {
                progress.Maximum += (int)(Math.Pow(3, i) / angles.Length);
            }
            if(len / minLen > 40)
            {
                progress.Maximum *= (int)(Math.Log(len, minLen) / 2);
            }
        }
        else if (angles.Length == 4)
        {
            for (int i = 0; i < count; i++)
            {
                progress.Maximum += (int)((Math.Pow(4, i) / angles.Length));
            }
            if ((double)len / minLen <= len * 0.05)
            {
                progress.Maximum *= (int)(Math.Pow(Math.Log(len, minLen),
1.4));
            }
        }
    }));
}

int x1, y1;
x1 = (int)(x + len * Math.Sin((2 * Math.PI * angle) / 360.0));
y1 = (int)(y + len * Math.Cos((2 * Math.PI * angle) / 360.0));
g.DrawLine(new Pen(pixels[i * pixels.Count * 0.1].Color, BranchWidth), x,
picture.Height - y, x1, picture.Height - y1);
i++;
if (len > minLen)
{
    for (int j = 0; j < angles.Length; j++)
    {
        DrawFractalTree(x1, y1, (int)(len / 1.5), angle + angles[j],
progress);
    }
}

progress.Invoke(new Action(() => // делегат для зміни progressBar
{

```

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				46
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        progress.PerformStep();
    }
}

}

public override void Info(TextBox info)
{
    info.Text = "Pythagoras, proving his famous theorem," +
        " built a figure with squares on the sides of a right-angled triangle." +
        " In our century, this figure of Pythagoras has grown into a whole tree."

+
    Environment.NewLine +
    Environment.NewLine +
    " For the first time, the Pythagorean tree was built" +
    " by A.E. Bosman (1891-1961) during the Second World War," +
    " using an ordinary drawing ruler. If in the classical Pythagorean" +
    " tree the angle is 45 degrees, then it is also possible to build" +
    " a generalized Pythagorean tree using other angles. Such a tree" +
    " is often called the windswept Pythagoras tree. If you draw only" +
    " the segments connecting the selected centers of the triangles in" +
    " some way, you get a naked Pythagorean tree." +
    Environment.NewLine +
    Environment.NewLine +
    "Angles (one angle - one branch)" +
    " (-35 - left branch will have angle 35)" +
    " or (45 right branch will have angle 45)" +
    " (e.g. (-35, 40, 10) we will have 3 branches" +
    " with corresponding angles)";

}

~FractalTree()
{
    Console.WriteLine("Class FractalTree is cleared");
}

}
}

```

### Class Barnsley\_fern.cs :

```

using System;
using System.Collections.Generic;
using System.Drawing;
using System.Windows.Forms;
using HelperClasses;

namespace FractalClasses
{
    public class Barnsley_fern:Fractals
    {
        protected float minX;
        protected float maxX;
        protected float minY;
        protected float maxY;
        protected int NumberOfPoints;
        protected float[] probability;
        protected float[,] Coefficient;
        protected int LenGradient;

        public Barnsley_fern(Bitmap picture, float maxX, float maxY, int NumberOfPoints,
            float[] probability, float[,] Coefficient, List<Pixel> pixels, Color backgroundColor, int
            LenGradient, float MinX = -3, float MinY = 0.1f):base(picture, pixels, backgroundColor)
    }
}

```

		Кормиш Р.І				Арк.
		Власенко О.В			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	47
Змн.	Арк.	№ докум.	Підпис	Дата		

```

{
    this.maxX = maxX;
    this.maxY = maxY;
    this.minX = MinX;
    this.minY = MinY;
    this.NumberOfPoints = NumberOfPoints;
    this.probability = probability;
    this.Coefficient = Coefficient;
    this.LenGradient = LenGradient;
}

public Barnsley_fern():base()
{

}

public Bitmap DrawBransleyFern(ProgressBar progress)
{
    Random random = new Random();
    float x0 = 0, y0 = 0;
    float x, y;
    int width = (int)(picture.Width / (maxX - minX));
    int height = (int)(picture.Height / (maxY - minY));
    int FunctionIndex = 0;

    base.Effects(g, BackgroundColor);

    progress.Invoke(new Action(() =>
    {
        progress.Maximum = NumberOfPoints;
        progress.Step = (NumberOfPoints / 100);
    }));

    for (int i = 1; i <= NumberOfPoints; i++)
    {
        // генерація числа (0;1)
        double randomNum = random.NextDouble();
        for (int j = 0; j < probability.Length; j++)
        {
            randomNum -= probability[j];
            if (randomNum <= 0)
            {
                FunctionIndex = j;
                break;
            }
        }
        // вичислення координат
        x = Coefficient[FunctionIndex, 0] * x0 + Coefficient[FunctionIndex, 1] *
y0 + Coefficient[FunctionIndex, 4];
        y = Coefficient[FunctionIndex, 2] * x0 + Coefficient[FunctionIndex, 3] *
y0 + Coefficient[FunctionIndex, 5];

        x0 = x;
        y0 = y;
        // перерахунок пікселів відносно форми
        x = (int)(x0 * width + picture.Width / 2);
        y = (int)(y0 * height);

        picture.SetPixel(Math.Abs((int)x) % picture.Width,
(int)(Math.Abs(picture.Height - (int)(y)) % picture.Height),

pixels[(int)((x * pixels.Count / LenGradient / (picture.Width *
0.00195))) % pixels.Count)).Color); // розтяг градієнта на весь папоротник
        if(i % (progress.Step) == 0) {

```

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				48
Змн.	Арк.	№ докум.	Підпис	Дата		



```

        progress.Invoke(new Action(() =>
        {
            progress.PerformStep();
        }));
    }
}
return picture;
}

public override void Info(TextBox info)
{
    info.Text = "The fern is one of the basic examples" +
        " of self-similar sets, i.e. it is a mathematically" +
        " generated pattern that can be reproducible at any" +
        " magnification or reduction. Like the Sierpinski triangle," +
        " the Barnsley fern shows how graphically beautiful structures" +
        " can be built from repetitive uses of mathematical formulas" +
        " with computers. " +
        Environment.NewLine +
        Environment.NewLine +
        "The fern code developed by Barnsley is an" +
        " example of an iterated function system(IFS) to create a fractal." +
        " This follows from the collage theorem.He has used fractals to" +
        " model a diverse range of phenomena in science and technology," +
        " but most specifically plant structures." +
        Environment.NewLine +
        Environment.NewLine +
        "IFSs provide models" +
        " for certain plants, leaves, and ferns, by virtue of the self" +
        " - similarity which often occurs in branching structures in nature." +
        "But nature also exhibits randomness and variation from one level" +
        " to the next; no two ferns are exactly alike," +
        " and the branching fronds become leaves at a smaller scale.";
}

~Barnsley_fern()
{
    Console.WriteLine("Class Barsley_fern is cleared");
}
}
}

```

### Class CurveDragon.cs :

```

using System;
using System.Drawing;
using System.Windows.Forms;

namespace FractalClasses
{
    public class CurveDragon : Fractals
    {
        private int i = 0;

        public CurveDragon(Bitmap picture, Color BackgroundColor):base(picture,
BackgroundColor)
        {
        }

        public CurveDragon():base(){ }
        public void DrawCurveDragon(int x1, int y1, int x2, int y2, int Iterations, Pen
pen, ProgressBar progress)
        {
            int NextX, NextY;

```

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				49
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        if (i == 0)
        {
            base.Effects(g, BackgroundColor);
        }

        if (Iterations == 0)
        {
            g.DrawLine(pen, x1, y1, x2, y2);

        }
        else if(Iterations > 0)
        {
            i++;
            NextX = (int)((x1 + x2) / 2 + (y2 - y1) / 2);
            NextY = (int)((y1 + y2) / 2 - (x2 - x1) / 2);

            DrawCurveDragon(x1, y1, NextX, NextY, Iterations - 1, pen, progress);
            DrawCurveDragon(x2, y2, NextX, NextY, Iterations - 1, pen, progress);

            //progress.Invoke(new Action(() => {
            //    progress.PerformStep();
            //}));
        }
    }

    public override void Info(TextBox info)
    {
        info.Text = "The Harter dragon, also known as" +
            " the Harter-Haytway dragon, was first explored" +
            " by NASA physicists John Heighway, Bruce Banks," +
            " and William Harter. It was described in 1967 by" +
            " Martin Gardner in the Math Games column of Scientific" +
            " American. Many of the properties of a fractal have been" +
            " described by Chandler Davis and Donald Knuth." +
            " The fractal can be created manually. " +
            Environment.NewLine +
            Environment.NewLine +
            "A dragon curve is any member of a family of self-similar" +
            " fractal curves, which can be approximated by recursive" +
            " methods such as Lindenmayer systems. The dragon curve" +
            " is probably most commonly thought of as the shape that" +
            " is generated from repeatedly folding a strip of paper in half," +
            " although there are other curves that are called dragon curves" +
            " that are generated differently." +
            Environment.NewLine +
            Environment.NewLine +
            "For this, we take a segment, bend it in half." +
            " Then we iterate over and over again. If we then" +
            " unbend the resulting (folded) line again so that all" +
            " angles are equal to 90 °, we get a dragon curve.";
    }

    ~CurveDragon()
    {
        Console.WriteLine("Class CurveDragon is cleared");
    }
}

```

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				50
Змн.	Арк.	№ докум.	Підпис	Дата		

## Бібліотека класів HelperClasses:

### Class Complex.cs :

```
using System;

namespace HelperClasses
{
    public class Complex
    {
        public double Re;
        public double Im;

        public Complex(double a, double b)
        {
            this.Re = a;
            this.Im = b;
        }

        public void Sqr()
        {
            double tmp = (Re * Re) - (Im * Im);
            Im = 2.0d * Re * Im;
            Re = tmp;
        }

        public double Magn()
        {
            return Math.Sqrt((Re * Re) + (Im * Im));
        }

        public void Add(Complex c)
        {
            Re += c.Re;
            Im += c.Im;
        }
    }
}
```

### Class Pixel.cs :

```
using System.Drawing;

namespace HelperClasses
{
    public class Pixel
    {
        public Color Color { get; set; }
    }
}
```

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				51
Змн.	Арк.	№ докум.	Підпис	Дата		

## Форми FractalCreator:

### GradientForm.cs:

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Windows.Forms;
using HelperClasses;

namespace FractalsCreator
{
    public partial class GradientForm : Form
    {
        internal List<Pixel> pixels;
        internal GradientForm()
        {
            InitializeComponent();
            TopMost = true;
            Focus();
            this.DoubleBuffered = true;
        }

        private void GradientForm_Load(object sender, EventArgs e)
        {
            ///
            ///
            /// Gradient
            ///
            ///

            // getting gradient pixels
            internal List<Pixel> GetPixels(Bitmap bitmap)
            {
                pixels = new List<Pixel>();
                for (int x = 0; x < bitmap.Width; x++)
                {
                    pixels.Add(new Pixel()
                    {
                        Color = bitmap.GetPixel(x, bitmap.Height / 2) // отримання кольору
                        ряду пікселів з градієнту
                    });
                }

                return pixels;
            }

            // random gradient creation
            internal void Gradient(PictureBox pictureBox)
            {
                Random color = new Random();

                int r = color.Next(256);
                int g = color.Next(256);
                int b = color.Next(256);
                int changer = color.Next(256);
                Bitmap Gradient = new Bitmap(pictureBox.Width, pictureBox.Height);
                switch (changer % 3)
```

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				52
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        {
            case 0:
            {
                SetGradient(Gradient, r, 255 - g, b, 2);
                break;
            }
            case 1:
            {
                SetGradient(Gradient, 255 - r, g, b, 1);
                break;
            }
            case 2:
            {
                SetGradient(Gradient, r, g, 255 - b, 0);
                break;
            }
            default:
                break;
        }

        pictureBox.Image = Gradient;
    }

    // generation gradient
    private void SetGradient(Bitmap Gradient, int r, int g, int b, int NoChangeIndex)
    {
        int[] change = new int[3];
        change[NoChangeIndex] = 0;
        for (int x = 0; x < Gradient.Width; x++)
        {
            for(int i = 0; i < change.Length; i++)
            {
                if(i != NoChangeIndex)
                {
                    change[i] = x;
                }
            }
            for (int y = 0; y < Gradient.Height; y++)
            {
                Gradient.SetPixel(x, y, Color.FromArgb(((r + change[0]) / 3) % 255,
                ((g + change[1]) / 3) % 255, ((b + change[2]) / 3) % 255));
            }
        }
    }

    // Random gradient button
    private void RandomGradient_Click(object sender, EventArgs e)
    {
        Gradient(pictureGradient);
    }

    // Gradient hot key
    private void GradientForm_KeyDown(object sender, KeyEventArgs e)
    {
        // generate gradient
        if (e.Control && e.KeyCode == Keys.Z)
        {
            RandomGradient.PerformClick();
        }
    }
    ~GradientForm()
    {
        Console.WriteLine("Form GradientForm is cleared");
    }
}

```

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				53
Змн.	Арк.	№ докум.	Підпис	Дата		

```
}
}
```

## FractalForm.cs:

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Globalization;
using System.Threading;
using FractalClasses;
using HelperClasses;

namespace FractalsCreator
{
    public partial class FractalForm : Form
    {
        ///
        ///
        ///     Start values
        ///
        ///

        internal Graphics g;
        internal Color BackgroundColor = Color.Transparent;
        internal List<Pixel> pixels;
        internal readonly GradientForm gradientForm;

        // start values for MBrot
        internal double hx, hy, maxZ, x_, y_;
        internal double ZoomVal = 1;

        internal Size size;
        internal double SizeArea;

        internal DateTime start, end;

        internal int selectFractal = 0;

        // start values for fractal tree
        internal List<NumericUpDown> Angles;

        // start values for curve dragon
        internal List<List<NumericUpDown>> Points;

        // list of the changing values
        internal readonly List<NumericUpDown> changingValues;
        NumericUpDown change;

        // information about fractals;
        internal Fractals info;

        // FractalForm FullScreen settings
        internal bool FullScreen = false;

        ///
        ///
        ///     Main form properties
        ///
    }
}
```

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				54
Змн.	Арк.	№ докум.	Підпис	Дата		

```

///

public FractalForm()
{
    InitializeComponent();
    TopMost = false;

    CultureInfo customCulture = (CultureInfo)
        Thread.CurrentThread.CurrentCulture.Clone();
    customCulture.NumberFormat.NumberDecimalSeparator = ".";
    Thread.CurrentThread.CurrentCulture = customCulture;

    image.Image = new Bitmap(image.Width, image.Height);
    size = image.Size;
    g = Graphics.FromImage(image.Image);
    this.DoubleBuffered = true;
    gradientForm = new GradientForm();

    // list of the curves dragon points
    Points = new List<List<NumericUpDown>>()
    {
        new List<NumericUpDown>{ FirstStartPointX, FirstStartPointY,
FirstEndPointX, FirstEndPointY },
        new List<NumericUpDown>{ SecondStartPointX, SecondStartPointY,
SecondEndPointX, SecondEndPointY },
        new List<NumericUpDown>{ ThirdStartPointX, ThirdStartPointY,
ThirdEndPointX, ThirdEndPointY },
        new List<NumericUpDown>{ FourthStartPointX, FourthStartPointY,
FourthEndPointX, FourthEndPointY },
        new List<NumericUpDown>{ FifthStartPointX, FifthStartPointY,
FifthEndPointX, FifthEndPointY },
        new List<NumericUpDown>{ SixthStartPointX, SixthStartPointY,
SixthEndPointX, SixthEndPointY },
        new List<NumericUpDown>{ SeventhStartPointX, SeventhStartPointY,
SeventhEndPointX, SeventhEndPointY },
        new List<NumericUpDown>{ EighthStartPointX, EighthStartPointY,
EighthEndPointX, EighthEndPointY }
    };

    // list of the angles
    Angles = new List<NumericUpDown> {
        FirstAngle, SecondAngle, ThirdAngle, FourthAngle
    };

    // list of the changing values
    changingValues = new List<NumericUpDown>
    {
        Iterations, BranchLength, NumberPoints, DragonIterations
    };
}

private void Form1_Load(object sender, EventArgs e)
{
    FractalsList.SelectedIndex = 0;
    ShowImageHeight.Value = image.Height;
    ShowImageWidth.Value = image.Width;

    // change the coordinates of the elements
    for (int i = 1; i < Angles.Count; i++)

```

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				55
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        {
            if (i == 1)
            {
                Angles[i].Location = new Point(Angles[i - 1].Location.X + 132,
Angles[i - 1].Location.Y);
            }
            else
            {
                Angles[i].Location = new Point(Angles[i - 2].Location.X, Angles[i -
2].Location.Y + 35);
            }
        }

        labelNumberOfCurves.Location = labelZOOM.Location;
        NumberOfCurves.Location = new Point(labelNumberOfCurves.Location.X +
labelNumberOfCurves.Width + 30, labelNumberOfCurves.Location.Y);

        LabelBranchLength.Location = labelZOOM.Location;
        BranchLength.Location = ZOOMValue.Location;

        labelStartX.Location = y.Location;
        labelStartY.Location = LabelMaxZDegreeTwo.Location;

        labelBranchWidth.Location = labelIterations.Location;
        BranchWidth.Location = Iterations.Location;

        StartX.Location = CenterY.Location;
        StartY.Location = MaxZDegreeTwo.Location;

        labelHorizontal.Location = y.Location;
        Horizontal.Location = CenterY.Location;

        labelVertical.Location = LabelMaxZDegreeTwo.Location;
        Vertical.Location = MaxZDegreeTwo.Location;

        labelNumberPoints.Location = labelIterations.Location;
        NumberPoints.Location = Iterations.Location;

        labelStartPoint.Location = new Point(labelNumberOfCurves.Location.X,
labelNumberOfCurves.Location.Y + 28);
        labelEndPoint.Location = new Point(labelStartPoint.Location.X +
labelStartPoint.Width + 30, labelStartPoint.Location.Y);

        labelStartPointX.Location = new Point(labelStartPoint.Location.X,
labelStartPoint.Location.Y + 23);
        labelStartPointY.Location = new Point(labelStartPointX.Location.X +
labelStartPointX.Width + 25, labelStartPointX.Location.Y);
        labelEndPointX.Location = new Point(labelEndPoint.Location.X,
labelEndPoint.Location.Y + 23);
        labelEndPointY.Location = new Point(labelEndPointX.Location.X +
labelEndPointX.Width + 25, labelEndPointX.Location.Y);

        FirstStartPointX.Location = new Point(labelStartPointX.Location.X,
labelStartPointX.Location.Y + 23);
        FirstStartPointY.Location = new Point(labelStartPointY.Location.X + 5,
labelStartPointY.Location.Y + 23);
        FirstEndPointX.Location = new Point(labelEndPointX.Location.X,
labelEndPointX.Location.Y + 23);
        FirstEndPointY.Location = new Point(labelEndPointY.Location.X + 5,
labelEndPointY.Location.Y + 23);

        for (int i = 1; i < Points.Count; i++)
        {
            for (int j = 0; j < 4; j++)

```

		Кормиш Р.І			ДУ «Житомирська політехніка». 21.121.12.000 - ПЗ	Арк.
		Власенко О.В				56
Змн.	Арк.	№ докум.	Підпис	Дата		



```

        {
            Points[i][j].Location = new Point(Points[i - 1][j].Location.X,
Points[i - 1][j].Location.Y + 28);
        }
    }

    labelDragonBrashWidth.Location = LabelMaxZDegreeTwo.Location;
    DragonBrashWidth.Location = MaxZDegreeTwo.Location;

    labelDragonIterations.Location = labelIterations.Location;
    DragonIterations.Location = Iterations.Location;

}

private void FractalForm_Resize(object sender, EventArgs e)
{
    if (FractalsList.SelectedIndex == 1)
    {
        if (StartX.Value > StartX.Minimum && StartX.Value < StartX.Maximum) {
StartX.Value += (image.Width - size.Width) / 2; };
        if (StartY.Value > StartY.Minimum && StartY.Value < StartY.Maximum) {
StartY.Value += (image.Height - size.Height) / 2; };
    }
    else if (FractalsList.SelectedIndex == 3)
    {
        for (int i = 0; i < Points.Count; i++)
        {
            for (int j = 0; j < Points[i].Count; j++)
            {
                if (Points[i][j].Value > Points[i][j].Minimum &&
Points[i][j].Value < Points[i][j].Maximum)
                {
                    if (j % 2 == 0)
                    {
                        Points[i][j].Value += (image.Width - size.Width) / 2;
                    }
                    else
                    {
                        Points[i][j].Value += (image.Height - size.Height) / 2;
                    }
                }
            }
        }
    }

    if (FractalsList.SelectedIndex != 0)
    {
        if (BackgroundColor == Color.Transparent)
        {
            image.BackColor = Color.White;
        }
        else
        {
            image.BackColor = BackgroundColor;
        }
    }
    size = image.Size;
    ShowImageHeight.Value = image.Height;
    ShowImageWidth.Value = image.Width;
}

private void FractalForm_ResizeEnd(object sender, EventArgs e)

```

		Кормиш Р.І			ДУ «Житомирська політехніка». 21.121.12.000 - ПЗ	Арк.
		Власенко О.В				57
Змн.	Арк.	№ докум.	Підпис	Дата		

```

{
    if (Progress.Value == Progress.Maximum)
    {
        DrawFractals();
    }
}
private void FractalForm_KeyDown(object sender, KeyEventArgs e)
{
    if (Progress.Value == Progress.Maximum)
    {
        change = changingValues[FractalsList.SelectedIndex];

        // save as
        if (e.Control && e.Shift && e.KeyCode == Keys.S)
        {
            SaveAs.PerformClick();
        }
        // load gradient
        else if (e.Control && e.Shift && e.KeyCode == Keys.Z)
        {
            LoadGradient.PerformClick();
        }
        // generate gradient
        else if (e.Control && e.KeyCode == Keys.Z)
        {
            GenerateGradientToolStripMenuItem.PerformClick();
        }
        // create or update fractal
        else if (e.Control && e.Shift && e.KeyCode == Keys.F)
        {
            if (tabControl.SelectedIndex == 2)
            {
                ButtonUpdate.PerformClick();
            }
            else
            {
                CreateFractal.PerformClick();
            }
        }
        // previous fractal
        else if (e.Shift && e.KeyCode == Keys.D1)
        {
            PreviousFractalToolStripMenuItem.PerformClick();
        }
        // next fractal
        else if (e.Shift && e.KeyCode == Keys.D2)
        {
            NextFractalToolStripMenuItem.PerformClick();
        }
        // Open GradientForm
        else if (e.Control && e.Shift && e.KeyCode == Keys.X)
        {
            GradientToolMenuItem.PerformClick();
        }
        else if (e.KeyCode == Keys.Oemplus && e.Shift)
        {
            IncreaseToolStripMenuItem.PerformClick();
        }
        else if (e.KeyCode == Keys.OemMinus && e.Shift)
        {
            DecreaseToolStripMenuItem.PerformClick();
        }
    }
}

```

		Кормиш Р.І			ДУ «Житомирська політехніка». 21.121.12.000 - ПЗ	Арк.
		Власенко О.В				58
Змн.	Арк.	№ докум.	Підпис	Дата		

```

// control keys
else if (FractalsList.SelectedIndex == 0)
{
    if (e.Shift && e.KeyCode == Keys.Q)
    {
        IncreaseTheZoomValueToolStripMenuItem.PerformClick();
    }
    else if (e.Shift && e.KeyCode == Keys.E)
    {
        DecreaseTheZoomValueToolStripMenuItem.PerformClick();
    }
    else if (e.KeyCode == Keys.Q)
    {
        IncreaseZOOM.PerformClick();
    }
    else if (e.KeyCode == Keys.E)
    {
        DecreaseZOOM.PerformClick();
    }
}

}

}

///
///
///     Tool Strip
///
///

// Create or update a fractal
private void CreateAndUpdateFractalToolStripMenuItem_Click(object sender,
EventArgs e)
{
    CreateFractal.PerformClick();
}

// Open gradient window
private async void Gradient_Click(object sender, EventArgs e)
{
    if (gradientForm.Visible == false)
    {
        await Task.Run(() => {

            gradientForm.ShowDialog();
            gradientForm.Focus();

        });
    }
}

// Generate Gradient
private void GenerateGradientToolStripMenuItem_Click(object sender, EventArgs e)
{
    gradientForm.Gradient(gradientForm.pictureGradient);
}

// Zoom in
private void ZoomInMBrotToolStripMenuItem_Click(object sender, EventArgs e)
{
    IncreaseZOOM.PerformClick();
}

```

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				59
Змн.	Арк.	№ докум.	Підпис	Дата		

```

// Zoom out
private void ZoomOutMBrotToolStripMenuItem_Click(object sender, EventArgs e)
{
    DecreaseZOOM.PerformClick();
}

// Increase the zoom value
private void IncreaseZoomValueToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (ZOOMValue.Value < ZOOMValue.Maximum)
    {
        ZOOMValue.Value += ZOOMValue.Increment;
    }
}

// Decrease the zoom value
private void DecreaseTheZoomValueToolStripMenuItem_Click(object sender, EventArgs
e)
{
    if (ZOOMValue.Value > ZOOMValue.Minimum)
    {
        ZOOMValue.Value -= ZOOMValue.Increment;
    }
}

// increase the most important value
private void IncreaseToolStripMenuItem_Click(object sender, EventArgs e)
{
    change = changingValues[FractalsList.SelectedIndex];

    if (change.Value < change.Maximum)
    {
        change.Value += change.Increment;
    }
}

// decrease the most important value
private void DecreaseToolStripMenuItem_Click(object sender, EventArgs e)
{
    change = changingValues[FractalsList.SelectedIndex];

    if (change.Value > change.Minimum)
    {
        change.Value -= change.Increment;
    }
}

// next fractal
private void NextFractalToolStripMenuItem_Click(object sender, EventArgs e)
{
    selectFractal += 1;
    FractalsList.SelectedIndex = selectFractal % 4;
}

// previous fractal
private void PreviousFractalToolStripMenuItem_Click(object sender, EventArgs e)
{
    selectFractal -= 1;
    if (selectFractal == -1)
    {
        selectFractal = FractalsList.Items.Count - 1;
    }
    FractalsList.SelectedIndex = selectFractal % 4;
}

```

		Кормиш Р.І			ДУ «Житомирська політехніка». 21.121.12.000 - ПЗ	Арк.
		Власенко О.В				60
Змн.	Арк.	№ докум.	Підпис	Дата		

```

///
///
///     Resize image
///
///

private void ShowImageHeight_ValueChanged(object sender, EventArgs e)
{
    this.Height += ((int)ShowImageHeight.Value - image.Height);
    image.Height = (int)ShowImageHeight.Value;
}
private void ShowImageWidth_ValueChanged(object sender, EventArgs e)
{
    this.Width += ((int)ShowImageWidth.Value - image.Width);
    image.Width = (int)ShowImageWidth.Value;
}

// FullScreen control
private void FractalForm_SizeChanged(object sender, EventArgs e)
{
    if (this.WindowState == FormWindowState.Maximized)
    {
        CreateFractal.PerformClick();
        FullScreen = true;
    }
    else if (this.WindowState == FormWindowState.Normal && FullScreen != false)
    {
        FullScreen = false;
        CreateFractal.PerformClick();
    }
}

///
///
///     UserDialogs
///
///

// Load Gradient
private void LoadGradient_Click(object sender, EventArgs e)
{
    OpenFileDialog.Title = "Load Gradient";
    if (OpenFileDialog.ShowDialog() == DialogResult.OK)
    {
        Bitmap isSufficientSize = new Bitmap(OpenFileDialog.FileName);
        int minimumImageWidth = 100; // minimum optimal pictures width for
gradient to avoid some problems with division by 0
        if (isSufficientSize.Width >= minimumImageWidth)
        {
            gradientForm.pictureGradient.Image = isSufficientSize;
            MessageBox.Show("Gradient loaded successfully", "Success",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        else
        {
            MessageBox.Show($"Image size is too small!\nMinimum image width :
{minimumImageWidth}\nCurrent width: {isSufficientSize.Width}", "Warning! Image size is
too small", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        }
    }
}
}

```

		Кормиш Р.І			ДУ «Житомирська політехніка». 21.121.12.000 - ПЗ	Арк.
		Власенко О.В				61
Змн.	Арк.	№ док.ум.	Підпис	Дата		

```

// Save image
private void SaveAs_Click(object sender, EventArgs e)
{
    if (image.Image != null)
    {
        SaveFileDialog SaveAs = new SaveFileDialog();
        SaveAs.Title = "Save as";
        SaveAs.OverwritePrompt = true;
        SaveAs.Filter = "(*.png)|*.png|(*.jpg)|*.jpg|(*.bmp)| *.bmp|All
files(*.*)|*.*";
        SaveAs.ShowHelp = true;
        if (SaveAs.ShowDialog() == DialogResult.OK)
        {
            try
            {
                image.Image.Save(SaveAs.FileName);
                MessageBox.Show("Fractal saved successfully", "Success",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
            catch
            {
                MessageBox.Show("Image can't be saved!!!", "Error!",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }
}

// selection of the background color for the image
private void ColorButton_Click(object sender, EventArgs e)
{
    ColorDialog BackColor = new ColorDialog();
    BackColor.AllowFullOpen = true;
    BackColor.FullOpen = true;
    BackColor.ShowHelp = true;
    BackColor.Color = Color.FromArgb(117, 238, 138);
    BackColor.AnyColor = true;
    BackColor.CustomColors = new int[]
    {
        636125, 382980, 5863935, 5427317, 1566114
    };
    if (BackColor.ShowDialog() == DialogResult.OK)
    {
        BackgroundColor = BackColor.Color;
        ColorButton.BackColor = BackColor.Color;
    }
}

///
///
///     Fractals selection and generation
///
///

// selection of the type of fractal
private void FractalsList_SelectedIndexChanged(object sender, EventArgs e)
{
    image.Image = null;
    image.BackColor = Color.White;
    PointsVisibleTrue(Points);
    AnglesVisibleTrue(Angles);
    FractalsInfo.Text = null;

    CulculationTime.Text = null;
}

```

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				62
Змн.	Арк.	№ докум.	Підпис	Дата		

```

IncreaseToolStripMenuItem.Text = null;
DecreaseToolStripMenuItem.Text = null;

Progress.Step = 1;

labelZOOM.Visible = false;
ZOOMValue.Visible = false;

DecreaseZOOM.Visible = false;
IncreaseZOOM.Visible = false;

ZoomNUM.Visible = false;

x.Visible = false;
y.Visible = false;

CenterX.Visible = false;
CenterY.Visible = false;

LabelMaxZDegreeTwo.Visible = false;
MaxZDegreeTwo.Visible = false;

LabelBranchLength.Visible = false;
BranchLength.Visible = false;

labelMinimalLength.Visible = false;
MinBranchLength.Visible = false;

labelStartX.Visible = false;
labelStartY.Visible = false;

StartX.Visible = false;
StartY.Visible = false;

labelBranchWidth.Visible = false;
BranchWidth.Visible = false;

labelAngles.Visible = false;
NumberOfAngles.Visible = false;

labelBackColor.Visible = false;
ColorButton.Visible = false;

labelIterations.Visible = false;
Iterations.Visible = false;

labelHorizontal.Visible = false;
Horizontal.Visible = false;

labelVertical.Visible = false;
Vertical.Visible = false;

labelNumberPoints.Visible = false;
NumberPoints.Visible = false;

labelNumberOfCurves.Visible = false;
NumberOfCurves.Visible = false;

labelStartPoint.Visible = false;
labelEndPoint.Visible = false;

```

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				63
Змн.	Арк.	№ докум.	Підпис	Дата		

```

labelStartPointX.Visible = false;
labelStartPointY.Visible = false;
labelEndPointX.Visible = false;
labelEndPointY.Visible = false;

labelDragonBrashWidth.Visible = false;
DragonBrashWidth.Visible = false;

labelDragonIterations.Visible = false;
DragonIterations.Visible = false;

GroupMouseControl.Visible = false;

zoomInMBrotToolStripMenuItem.Visible = false;
zoomOutMBrotToolStripMenuItem.Visible = false;
IncreaseTheZoomValueToolStripMenuItem.Visible = false;
DecreaseTheZoomValueToolStripMenuItem.Visible = false;

BackgroundColor = Color.Transparent;
ColorButton.BackColor = Color.White;

if (FractalsList.SelectedIndex == 0)
{
    info = new MBrotSet();
    info.Info(FractalsInfo); // add info about the Mandelbrot set
    hx = -0.6;
    hy = 0;
    maxZ = 4;
    CenterX.Text = hx.ToString();
    CenterY.Text = hy.ToString();
    MaxZDegreeTwo.Text = maxZ.ToString();
    ZoomVal = 1;
    SizeArea = 4;

    image.Enabled = false;

    DecreaseZOOM.Enabled = false;
    IncreaseZOOM.Enabled = false;

    labelZOOM.Visible = true;
    ZOOMValue.Visible = true;

    DecreaseZOOM.Visible = true;
    IncreaseZOOM.Visible = true;

    x.Visible = true;
    y.Visible = true;

    CenterX.Visible = true;
    CenterY.Visible = true;

    LabelMaxZDegreeTwo.Visible = true;
    MaxZDegreeTwo.Visible = true;

    labelIterations.Visible = true;
    Iterations.Visible = true;

    GroupMouseControl.Visible = true;

    IncreaseToolStripMenuItem.Text = "Increase the value of the max
iterations";
    DecreaseToolStripMenuItem.Text = "Decrease the value of the max
iterations";

```

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				64
Змн.	Арк.	№ докум.	Підпис	Дата		



```

        zoomInMBrotToolStripMenuItem.Visible = true;
        zoomOutMBrotToolStripMenuItem.Visible = true;
        IncreaseTheZoomValueToolStripMenuItem.Visible = true;
        DecreaseTheZoomValueToolStripMenuItem.Visible = true;
    }
    else if (FractalsList.SelectedIndex == 1)
    {
        info = new FractalTree();
        info.Info(FractalsInfo); // add info about the fractal tree

        NumberOfAngles.SelectedIndex = 1;
        StartX.Value = image.Width / 2;
        StartY.Value = image.Height / 5;

        labelBackColor.Location = x.Location;
        ColorButton.Location = CenterX.Location;

        LabelBranchLength.Visible = true;
        BranchLength.Visible = true;

        labelMinimallLength.Visible = true;
        MinBranchLength.Visible = true;

        labelStartX.Visible = true;
        labelStartY.Visible = true;

        labelBranchWidth.Visible = true;
        BranchWidth.Visible = true;

        StartX.Visible = true;
        StartY.Visible = true;

        labelAngles.Visible = true;
        NumberOfAngles.Visible = true;

        AnglesVisibleTrue(Angles, NumberOfAngles.SelectedIndex);

        labelBackColor.Visible = true;
        ColorButton.Visible = true;

        IncreaseToolStripMenuItem.Text = "Increase the value of the branch
length";
        DecreaseToolStripMenuItem.Text = "Decrease the value of the branch
length";
    }
    else if (FractalsList.SelectedIndex == 2)
    {
        info = new Barnsley_fern();
        info.Info(FractalsInfo); // add info about the Barnsley fern

        labelBackColor.Location = x.Location;
        ColorButton.Location = CenterX.Location;

        labelBackColor.Visible = true;
        ColorButton.Visible = true;

        labelHorizontal.Visible = true;
        Horizontal.Visible = true;

        labelVertical.Visible = true;
        Vertical.Visible = true;
    }

```

		Кормиш Р.І			ДУ «Житомирська політехніка». 21.121.12.000 - ПЗ	Арк.
		Власенко О.В				65
Змн.	Арк.	№ докум.	Підпис	Дата		

```

labelNumberPoints.Visible = true;
NumberPoints.Visible = true;

IncreaseToolStripMenuItem.Text = "Increase the value of the Numbers
points";
DecreaseToolStripMenuItem.Text = "Decrease the value of the Numbers
points";
}
else if (FractalsList.SelectedIndex == 3)
{
    info = new CurveDragon();
    info.Info(FractalsInfo); // add info about the curve of dragon

    NumberOfCurves.SelectedIndex = 3;
    PointsVisibleTrue(Points, NumberOfCurves.SelectedIndex);

    // start point of the dragon curves
    for (int i = 0; i < Points.Count; i++)
    {
        if (i == 0)
        {
            Points[i][0].Value = image.Width / 2;
            Points[i][1].Value = image.Height / 2;
            Points[i][2].Value = image.Width / 2 - 300;
            Points[i][3].Value = image.Height / 2 - 200;
        }
        else
        {
            Points[i][0].Value = Points[i - 1][0].Value;
            Points[i][1].Value = Points[i - 1][1].Value;
            Points[i][2].Value = Points[i - 1][2].Value - 200;
            Points[i][3].Value = Points[i - 1][3].Value - 300;
        }
    }
    labelBackColor.Visible = true;
    labelBackColor.Location = y.Location;

    ColorButton.Visible = true;
    ColorButton.Location = CenterY.Location;

    Progress.Visible = true;

    labelNumberOfCurves.Visible = true;
    NumberOfCurves.Visible = true;

    labelStartPoint.Visible = true;
    labelEndPoint.Visible = true;

    labelStartPointX.Visible = true;
    labelStartPointY.Visible = true;
    labelEndPointX.Visible = true;
    labelEndPointY.Visible = true;

    labelDragonBrashWidth.Visible = true;
    DragonBrashWidth.Visible = true;

    labelDragonIterations.Visible = true;
    DragonIterations.Visible = true;

    IncreaseToolStripMenuItem.Text = "Increase the value of the Numbers
points";
    DecreaseToolStripMenuItem.Text = "Decrease the value of the Numbers
points";

```

		Кормиш Р.І			ДУ «Житомирська політехніка». 21.121.12.000 - ПЗ	Арк.
		Власенко О.В				66
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
    DrawFractals();
}
// button for generating a fractal
private void GenerateFractal_Click(object sender, EventArgs e)
{
    DrawFractals();
}

// drawing Fractals
internal void DrawFractals()
{
    Progress.Invoke(new Action(() => {
        Progress.Minimum = 0;
        Progress.Value = Progress.Minimum;
    }));

    pixels = gradientForm.GetPixels((Bitmap)gradientForm.pictureGradient.Image);
    size = image.Size;

    if (FractalsList.SelectedIndex == 0)
    {
        bool er_x, er_y, er_z;
        ZoomNUM.Text = ZoomVal.ToString("F0") + " X";
        ZoomNUM.Visible = true;
        image.Enabled = true;
        DecreaseZOOM.Enabled = true;
        IncreaseZOOM.Enabled = true;

        CenterX.ForeColor = Color.Black;
        CenterY.ForeColor = Color.Black;
        MaxZDegreeTwo.ForeColor = Color.Black;
        er_x = double.TryParse(CenterX.Text, out hx);
        er_y = double.TryParse(CenterY.Text, out hy);
        er_z = double.TryParse(MaxZDegreeTwo.Text, out maxZ);

        if (!er_x)
        {
            CenterX.ForeColor = Color.Red;
            MessageBox.Show("Error entering value x!", "Error",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        else if (!er_y)
        {
            CenterY.ForeColor = Color.Red;
            MessageBox.Show("Error entering value y!", "Error",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        else if (!er_z)
        {
            MaxZDegreeTwo.ForeColor = Color.Red;
            MessageBox.Show("Error entering value max |z| ^ 2!", "Error",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        else
        {
            DrawMBrot();
        }
    }
    else if (FractalsList.SelectedIndex == 1)

```

		Кормиш Р.І			ДУ «Житомирська політехніка». 21.121.12.000 - ПЗ	Арк.
		Власенко О.В				67
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        {
            image.Image = null;
            DrawFractalTree();
        }
        else if (FractalsList.SelectedIndex == 2)
        {
            image.Image = null;
            DrawBarnsleyFern();
        }
        else if (FractalsList.SelectedIndex == 3)
        {
            image.Image = null;
            DrawCurveDragon();
        }
    }

    ///
    ///
    ///     Mandelbrot set
    ///
    /// Draw MBrot set
    private async void DrawMBrot()
    {
        image.Cursor = Cursors.WaitCursor;
        image.Enabled = false;    ///
        IncreaseZOOM.Enabled = false;    ///
        DecreaseZOOM.Enabled = false;    /// disabled access to change the image
        CreateFractal.Enabled = false;    ///
        FractalsList.Enabled = false;    ///

        start = DateTime.Now;
        ZoomNUM.Width = (int)(ZoomNUM.Text.Length * ZoomNUM.Font.Height * 0.4);

        image.Invalidate();
        pixels = gradientForm.GetPixels((Bitmap)gradientForm.pictureGradient.Image);
        Bitmap pictureMBrotSet = new Bitmap(image.Width, image.Height);
        MBrotSet MBrotSet = new MBrotSet((Bitmap)pictureMBrotSet, pixels, size,
        gradientForm.pictureGradient, (int)Iterations.Value);
        int CountColors = 100;

        await Task.Run(() =>
        {
            pictureMBrotSet = MBrotSet.CalculationMBrot(hx, hy, x_, y_, maxZ,
            SizeArea, CountColors, Progress);
            image.Image = pictureMBrotSet;
        });

        end = DateTime.Now;
        CulculationTime.Text = (end - start).TotalMilliseconds.ToString("F2") + "
ms";

        image.Enabled = true;    ///
        IncreaseZOOM.Enabled = true;    ///
        DecreaseZOOM.Enabled = true;    /// added access to change the image
        CreateFractal.Enabled = true;    ///
        FractalsList.Enabled = true;    ///
        image.Cursor = Cursors.Default;
    }

```

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				68
Змн.	Арк.	№ докум.	Підпис	Дата		

```

// Control buttons for MBrot set
private void DecreaseZOOM_Click(object sender, EventArgs e)
{
    SizeArea *= (double)ZOOMValue.Value;
    CenterX.Text = hx.ToString();
    CenterY.Text = hy.ToString();
    ZoomVal /= (double)ZOOMValue.Value;
    ZoomNUM.Visible = true;
    ZoomNUM.Clear();
    ZoomNUM.Text = ZoomVal.ToString("F2") + " X";
    DrawFractals();
}
private void IncreaseZOOM_Click(object sender, EventArgs e)
{
    SizeArea /= (double)ZOOMValue.Value;
    CenterX.Text = hx.ToString();
    CenterY.Text = hy.ToString();
    ZoomVal *= (double)ZOOMValue.Value;
    ZoomNUM.Visible = true;
    ZoomNUM.Clear();
    if (ZoomVal < 1)
    {
        ZoomNUM.Text = ZoomVal.ToString("F4") + " X";
    }
    else
    {
        ZoomNUM.Text = ZoomVal.ToString("F2") + " X";
    }
    DrawFractals();
}
private void Image_MouseClick(object sender, MouseEventArgs e)
{
    size = image.Size;
    int X = e.X, Y = e.Y;
    if (FractalsList.SelectedItem == FractalsList.Items[0])
    {
        ZoomNUM.Visible = true;
        if (e.Button == MouseButtons.Left)
        {
            hx = (hx - SizeArea / 2) + X * (SizeArea / size.Width);
            hy = (hy - SizeArea / 2) + Y * (SizeArea / size.Height);
            SizeArea /= (double)ZOOMValue.Value; // increase zoom by ZOOMValue

            CenterX.Text = hx.ToString();
            CenterY.Text = hy.ToString();
            ZoomVal *= (double)ZOOMValue.Value;
            ZoomNUM.Clear();
            ZoomNUM.Text = ZoomVal.ToString("F2") + " X";
            DrawFractals();
        }
        else if (e.Button == MouseButtons.Middle)
        {
            // back to default
            SizeArea = 4;
            hx = -0.6;
            hy = 0;
            CenterX.Text = hx.ToString();
            CenterY.Text = hy.ToString();
            ZoomVal = 1;
            ZoomNUM.Clear();
            ZoomNUM.Width = 80;

            ZoomNUM.Text = ZoomVal.ToString("F2") + " X";
            DrawFractals();
        }
    }
}

```

value

		Кормиш Р.І			ДУ «Житомирська політехніка». 21.121.12.000 - ПЗ	Арк.
		Власенко О.В				69
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
    else if (e.Button == MouseButton.Right)
    {
        x_ = (hx - SizeArea / 2) + X * (SizeArea / size.Width);
        y_ = (hy - SizeArea / 2) + Y * (SizeArea / size.Height);
        SizeArea *= (double)ZOOMValue.Value; // decrease zoom by ZOOMValue

        CenterX.Text = x_.ToString();
        CenterY.Text = y_.ToString();
        ZoomVal /= (double)ZOOMValue.Value;
        ZoomNUM.Clear();
        if (ZoomVal < 1)
        {
            ZoomNUM.Text = ZoomVal.ToString("F4") + " X";
        }
        else
        {
            ZoomNUM.Text = ZoomVal.ToString("F2") + " X";
        }
        DrawFractals();
    }
}

///
///
///     Fractal tree
///
///

// selection of the number of angles
private void NumberOfAngles_SelectedIndexChanged(object sender, EventArgs e)
{
    AnglesVisibleTrue(Angles);
    AnglesVisibleTrue(Angles, NumberOfAngles.SelectedIndex);
}

// Draw fractal tree
private async void DrawFractalTree()
{
    image.Cursor = Cursors.WaitCursor;
    FractalsList.Enabled = false;
    CreateFractal.Enabled = false;
    start = DateTime.Now;
    image.Invalidate();

    Bitmap pictureTree = new Bitmap(image.Width, image.Height);
    int branchLength = (int)BranchLength.Value;

    double[] angles = new double[NumberOfAngles.SelectedIndex + 1];

    for (int i = 0; i < angles.Length; i++)
    {
        angles[i] = (double)Angles[i].Value;
    }

    FractalTree tree = new FractalTree(pictureTree, pixels, angles,
    (int)(MinBranchLength.Value), (int)(BranchWidth.Value), BackgroundColor);
    await Task.Run(() =>
    {
        tree.DrawFractalTree((int)(StartX.Value), (int)(StartY.Value),
        branchLength, 0, Progress);
    });
}

```

		Кормиш Р.І			ДУ «Житомирська політехніка». 21.121.12.000 - ПЗ	Арк.
		Власенко О.В				70
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        image.Image = pictureTree;

    });

    if (NumberOfAngles.SelectedIndex == 0)
    {
        Progress.Maximum = 10;
    }
    Progress.Value = Progress.Maximum;

    end = DateTime.Now;
    CulculationTime.Text = (end - start).TotalMilliseconds.ToString("F2") + "
ms";

    CreateFractal.Enabled = true;
    FractalsList.Enabled = true;
    image.Cursor = Cursors.Default;
}

// changing the number of visible branches
private void AnglesVisibleTrue(List<NumericUpDown> Angles, int endIndex = -1)
{
    if (endIndex == -1)
    {
        for (int i = 0; i < Angles.Count; i++)
        {
            Angles[i].Visible = false;
        }
    }
    else
    {
        for (int i = 0; i <= endIndex; i++)
        {
            Angles[i].Visible = true;
        }
    }
}

// calculation of the changing mininum value of the branch length
private void BranchLength_ValueChanged(object sender, EventArgs e)
{
    if (NumberOfAngles.SelectedIndex == 3)
    {
        MinBranchLength.Minimum = (int)(Math.Floor(BranchLength.Value / 40));
    }
    else if (NumberOfAngles.SelectedIndex == 2)
    {
        MinBranchLength.Minimum = (int)(Math.Floor(BranchLength.Value / 60));
    }
    else
    {
        MinBranchLength.Minimum = (int)(Math.Floor((double)BranchLength.Value *
0.005)) + 1;
    }

    if (MinBranchLength.Value < MinBranchLength.Minimum)
    {
        MinBranchLength.Value = MinBranchLength.Minimum;
    }
}
}

```

		Кормиш Р.І			ДУ «Житомирська політехніка». 21.121.12.000 - ПЗ	Арк.
		Власенко О.В				71
Змн.	Арк.	№ док.ум.	Підпис	Дата		

```

///
///
///     Fern Barnsley
///
///

// Draw fern Barnsley
private async void DrawBarnsleyFern()
{
    image.Cursor = Cursors.WaitCursor;
    start = DateTime.Now;
    image.Invalidate();
    CreateFractal.Enabled = false;
    FractalsList.Enabled = false;

    float[] probability = new float[4] { 0.01f, 0.06f, 0.08f, 0.85f };
    int NumberOfPoint = (int)NumberPoints.Value;
    float maxX = 3 + (float)Horizontal.Value, maxY = 10.1f +
(float)Vertical.Value;
    float[,] Coefficient = new float[4, 6]
    {
        {0, 0, 0, 0.16f, 0, 0},
        {-0.15f, 0.28f, 0.26f, 0.24f, 0, 0.44f},
        {0.2f, -0.26f, 0.23f, 0.22f, 0, 1.6f},
        {0.85f, 0.04f, -0.04f, 0.85f, 0, 1.6f}
    };

    Bitmap pictureFern = new Bitmap(image.Width, image.Height);
    Barnsley_fern Fern = new Barnsley_fern(pictureFern, maxX, maxY,
NumberOfPoint, probability, Coefficient, pixels, BackgroundColor,
gradientForm.pictureGradient.Width);
    await Task.Run(() =>
    {
        pictureFern = Fern.DrawBransleyFern(Progress);
        image.Image = pictureFern;
    });

    end = DateTime.Now;
    CulculationTime.Text = (end - start).TotalMilliseconds.ToString("F2") + "
ms";

    CreateFractal.Enabled = true;
    FractalsList.Enabled = true;
    image.Cursor = Cursors.Default;
}

///
///
///     Dragon curve
///
///

// Draw dragon curve
private async void DrawCurveDragon()
{
    image.Cursor = Cursors.WaitCursor;
    start = DateTime.Now;

    image.Invalidate();

    CreateFractal.Enabled = false;

```

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				72
Змн.	Арк.	№ докум.	Підпис	Дата		



```

FractalsList.Enabled = false;
int brushWidth = (int)(DragonBrushWidth.Value);
int CountIterations;
Bitmap pictureCurveDragon = new Bitmap(image.Width, image.Height);

List<int> indexes = new List<int>();
List<List<int>> Coords = new List<List<int>>();
List<int> coordinates;

for (int i = 0; i <= NumberOfCurves.SelectedIndex; i++)
{
    coordinates = new List<int>();
    for (int j = 0; j < 4; j++)
    {
        coordinates.Add((int)Points[i][j].Value);
    }
    Coords.Add(coordinates);
}

// obtaining color for coloring
for (int i = 0; i < Coords.Count; i++)
{
    indexes.Add((int)((i + 1) * gradientForm.pictureGradient.Image.Width /
(Coords.Count + 1)));
}

CurveDragon Dragon = new CurveDragon(pictureCurveDragon, BackgroundColor);
CountIterations = (int)(DragonIterations.Value);
Progress.Maximum = 0;
for (int i = CountIterations; i >= 0; i--)
{
    Progress.Maximum += (int)Math.Pow(2, i);
}
Progress.Maximum *= (Coords.Count);

await Task.Run(() => {
    for (int i = 0; i < Coords.Count; i++)
    {
        Dragon.DrawCurveDragon(Coords[i][0], image.Height - Coords[i][1],
Coords[i][2], image.Height - Coords[i][3], CountIterations, new
Pen(pixels[indexes[i]].Color, brushWidth), Progress);
    }
    image.Image = pictureCurveDragon;
    Progress.Invoke(new Action(() =>
    {
        Progress.Value = Progress.Maximum;
    }));
});

end = DateTime.Now;
CulculationTime.Text = (end - start).TotalMilliseconds.ToString("F2") + "
ms";

CreateFractal.Enabled = true;
FractalsList.Enabled = true;
image.Cursor = Cursors.Default;
}

```

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				73
Змн.	Арк.	№ докум.	Підпис	Дата		

```

// selection of the number of dragon curves
private void NumberOfCurves_SelectedIndexChanged(object sender, EventArgs e)
{
    PointsVisibleTrue(Points);
    PointsVisibleTrue(Points, NumberOfCurves.SelectedIndex);
}

// changing the number of visible curves
internal void PointsVisibleTrue(List<List<NumericUpDown>> Points, int endIndex =
-1)
{
    if (endIndex == -1)
    {
        for (int i = 0; i < Points.Count; i++)
        {
            for (int j = 0; j < Points[i].Count; j++)
            {
                Points[i][j].Visible = false;
            }
        }
    }
    else
    {
        for (int i = 0; i <= endIndex; i++)
        {
            for (int j = 0; j < Points[i].Count; j++)
            {
                Points[i][j].Visible = true;
            }
        }
    }
}

~FractalForm()
{
    Console.WriteLine("Class FractalForm is clear");
}
}

```

		Кормиш Р.І			ДУ «Житомирська політехніка».21.121.12.000 - ПЗ	Арк.
		Власенко О.В				74
Змн.	Арк.	№ докум.	Підпис	Дата		