|  |
| --- |
| ВШЭ |
| **Пояснительная записка** |
| **Программа для построения фракталов Кривая Гильберта, С-Кривая Леви, Множество Кантора** |
|  |
| **Герасименко Е.Р. БПИ-183 вариант 38** |
| **09.12.2018** |

|  |
| --- |
| [Введите аннотацию документа. Аннотация обычно представляет собой краткий обзор содержимого документа. Введите аннотацию документа. Аннотация обычно представляет собой краткий обзор содержимого документа.] |

**ТИТУЛЬНЫЙ ЛИСТ**

**Пояснительная записка**

**Программа для построения фракталов**

**Кривая Гильберта, С-Кривая Леви, Множество Кантора**

**Выполнил Герасименко Е.Р. БПИ-183 вариант 38**

ОГЛАВЛЕНИЕ

# 1. Условие задачи

Разработать оконное приложение (Windows Forms Application (изучается на семинарах и лекциях) или WPF (изучается самостоятельно)), позволяющее:

1. Отрисовывать три вида фракталов, которые определены в индивидуальном варианте. Вариант 38: Кривая Гильберта, С-Кривая Леви, Множество Кантора.
2. Предоставлять пользователю выбор текущего фрактала для отрисовки.
3. Предоставлять пользователю возможность устанавливать количество шагов

рекурсии (её глубину - количество рекурсивных вызовов). При изменении глубины рекурсии фрактал должен быть автоматически перерисован. Следите за переполнением стека.

1. Автоматически перерисовывать фрактал при изменении размеров окна. Окно обязательно должно быть масштабируемым. Вы можете задать минимальный и максимальный размер окна. Максимальным считается размер окна, соответствующий размеру экрана, а минимальным размером окна считается половинный размер экрана (как по длине, так и по ширине).
2. Предоставлять пользователю возможность выбора двух цветов startColor и

endColor. Цвет startColor используется для отрисовки элементов первой итерации,

цвет endColor - для отрисовки элементов последней итерации. Цвета для

промежуточных итераций должны вычисляться с использованием линейного

градиента. http://qaru.site/questions/222000/generate-color-gradient-in-c

Например,



Рис. 1. Градиент цвета

Желтый цвет используется для отрисовки элементов первой итерации. Синий - для

отрисовки элементов последней итерации.

Промежуточные цвета вычисляются исходя из начального и конечного значений

цвета и номера итерации (не генерируются случайным образом).

1. Сообщать о некорректном вводе данных, противоречивых или недопустимых

значениях данных и других нештатных ситуациях во всплывающих окнах типа

MessageBox.

1. Должна быть предусмотрена возможность сохранения фрактала в виде картинки

(формат выбрать самостоятельно).

1. Предусмотреть возможность изменения масштаба фрактала для его детального

просмотра. Увеличение должно быть 2, 3 и 5-кратным

1. Предусмотреть возможность перемещения изображения, в т.ч. при увеличенном

изображении (пункт 8).

1. На интерфейсе может быть предусмотрена дополнительная функциональность на

Ваше усмотрение.

Для того, чтобы выделить базовую структуру алгоритма построения фракталов следует разбить процедуру рисования на небольшие действия, и те из них, которые окажутся одинаковыми для разных фракталов, поместить в базовый (родительский) класс.

Различающиеся действия поместить в производные классы, создав по одному производному классу для каждого фрактала.

Определить в базовом классе Fractal следующие члены:

* поля:
  + длина отрезка (сторона квадрата, треугольника, радиус окружности и т.д.)
  + на первой итерации
  + startColor (см. п. 5)
  + endColor (см. п. 5)
  + текущий уровень рекурсии
  + максимальный уровень рекурсии, заданный пользователем.
* метод Draw (может быть абстрактным или виртуальным, зависит от архитектуры ваших классов)

В зависимости от типа фрактала в классы-наследники следует добавить необходимые члены класса, указанные при описании фрактала. Допустимо добавлять другие члены класса.

Ограничения:

1. В программной реализации не использовать вспомогательные компоненты и сторонние библиотеки, не входящие в стандартную библиотеку.  
2. Не использовать массивы типа object[].

# 2. Функции разрабатываемого приложения

# 2.1. Варианты использования

Программа предназначена для использования на компьютере, содержащем пакеты, необходимые для использования С# и .NET, в образовательных, развлекательных и научных целях требующих построения фракталов Кривая Гильберта, С-Кривая Леви, Множество Кантора.

# 2.2. Описание интерфейса пользователя

Интерфейс представляет из себя оконный интерфейс Windows Forms и содержит:

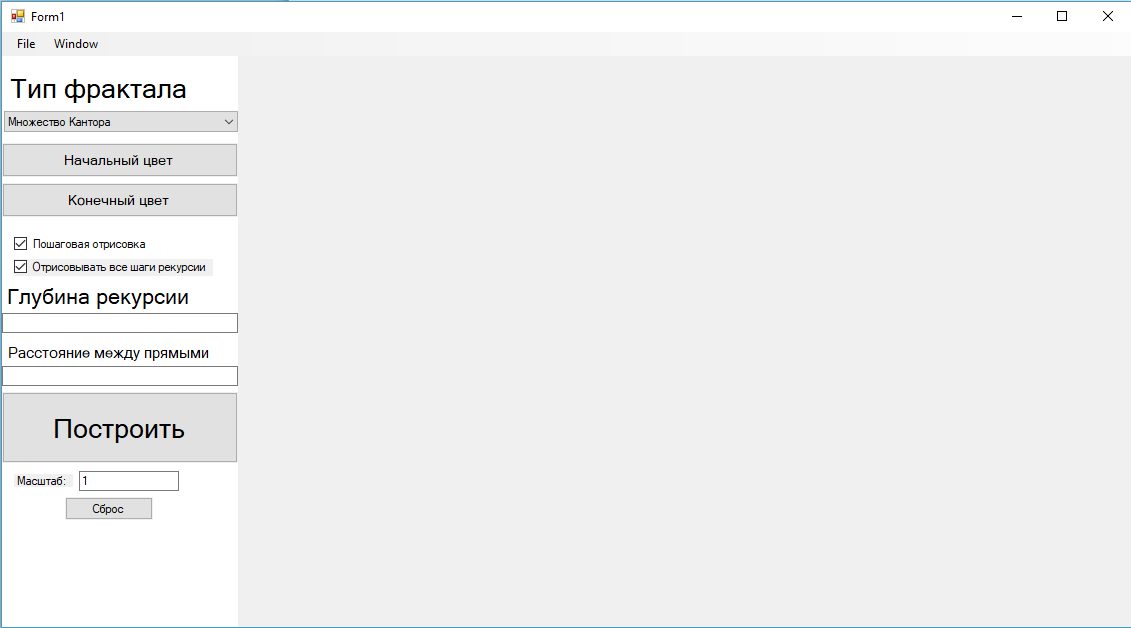


Рисунок 1 - Интерфейс программы

* Выпадающий список фракталов для выбора типа фрактала
* Кнопку для выбора начального цвета градиента
* Кнопку для выбора конечного цвета градиента
* Поле для выбора отрисовывать пошагово, или сразу всё
* Поле для выбора отрисовывать все шаги рекурсии или только видимые
* Поле для ввода глубины рекурсии
* Поле для ввода расстояния между элементами (Множество Кантора)
* Кнопка «Построить»
* Поле для ввода масштаба
* Кнопка «Сброс», которая устанавливает стартовую позицию и увеличение равное 1
* Меню File
  + Save
    - Save As – сохранить как - программа попросит выбрать путь и ввести имя
    - Save – сохранить - программа сохранит под указанным ранее именем, если имя не было у казано ранее – вызовется Save As
  + New - открывается новое окно фрактала, данное окно становится неактивным
* Меню Window
  + Over all windows – показывать окно поверх остальных (если галочка – да)

При отрисовке фрактала на экран выводится окно. Данное окно содержит:

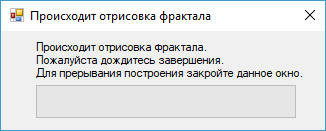


Рисунок 2 - Интерфейс окна отрисовки фрактала

* Текст «Происходит отрисовка фрактала. Пожалуйста дождитесь завершения. Для прерывания построения закройте данное окно.»
* Шкалу, отражающую, насколько завершено построение фрактала (далее - шкала прогресса).

# 3. Структура приложения

Класс:

* Fractal

Поля:

* + handle - используется для определения конца работы потока построения фрактала
  + \_vlock - используется для безопасного изменения значения количества вычисленных элементов фрактала
  + f - первая буква названия фрактала
  + drawall – поле отвечающее за то нужно ли строить все элементы фрактала (если истина) или нужно строить только те элементы, чьи линейные размеры по всем осям больше 0 пикселей (они будут изображены) (если ложь)
  + max\_length - количество элементов фрактала (при наличии в фрактале более 230 элементов max\_length = 230), после начала отрисовки шкалы прогресса принимает значение 10000
  + max\_length2 - количество элементов фрактала (при наличии в фрактале более 230 элементов max\_length = 230),
  + step - шаг заполнения шкалы прогресса, численно равен 1 части фрактала (1 часть фрактала соответствует max\_length/max\_length2, например если max\_length2=100000,(max\_length=10000),то 1 часть фрактала равна 0.1 (step=0.1) )
  + sum - поле отвечающее за то, на сколько заполнена шкала прогресса (при отрисовке каждого элемента фрактала увеличивается на step)
  + isdrawing - поле отвечающее за состояние процесса отрисовки фрактала (истина – происходит отрисовка, ложь - нет)
  + scale - коэффициент увеличения (приближения) фрактала
  + pb - ccылка на форму содержащую шкалу прогресса
  + pbm - максимальное значение шкалы прогресса
  + scf - установлен ли начальный цвет градиента (истина - установлен, ложь – нет)
  + ecf - установлен ли конечный цвет градиента (истина - установлен, ложь – нет)
  + colorarr - массив цветов (возможно вычисление цвета по ходу отрисовки элементов фрактала, но это усложняет реализацию отрисовки фрактала в параллельном потоке и убирает возможность последующего распараллеливания процесса отрисовки фрактала).
  + colorarrmax - длинна массива цветов
  + colorarriter - номер взятого на данной итерации элемента массива colorarr
  + colorarrstep - шаг при переходе на следующую итерацию (может требоваться для оптимизации используемой памяти при отрисовке более чем 256 рекурсий)
  + startColor - начальный цвет градиента
  + endColor - конечный цвет градиента
  + level\_of\_rec - текущий уровень рекурсии
  + max\_level\_of\_rec - максимальный уровень рекурсии
  + space - ширина полей от изображения фрактала, до границы закрашенной области
  + xspace - расстояние от левого края формы, до границы закрашенной области
  + yspace - расстояние от верхнего края формы, до границы закрашенной области
  + size - размеры квадрата ограничивающего фрактал
  + xsize - ширина фрактала
  + ysize - высота фрактала
  + (xleft,yleft) - координата левого верхнего угла области для рисования фрактала
  + pictureBoxXsize - ширина окна приложения
  + pictureBoxYsize - высота окна приложения
  + message - способ передачи информации о различных непредвиденных ситуациях в основное приложение из потока вычислений
  + pen - ручка для рисования линий
  + brush – кисть для заливки фигур

Свойства:

* + Max\_length - свойство для вычисления max\_length и max\_length2

Методы:

* + Конаструкторы - задаёт изначальные значения, затем те, которые поданы в него:
    - startColor = Color.White;
    - endColor = Color.White;
    - level\_of\_rec = 0;
    - max\_level\_of\_rec = 1;
    - size = 300;
    - space = 10;
    - xspace = 218;
    - yspace = 0;
    - xsize = ysize = 300;
  + binpow(x, step) - функция быстрого возведения x в степень step
  + bindrob(max,x,step) - функция точного деления max на x step раз (max/(x\*x\*...\*x))
  + setpictureBoxsize(x,y) - устанавливает pictureBoxXsize и pictureBoxYsize как x и y
  + set\_float(f) - устанавливает значение переменной = f (только множество Кантора)
  + Draw - метод, который запускает отрисовку фрактала
* Point

Поля:

* + x - координата по оси X
  + y - координата по оси Y

Методы:

* + Конструктор - задаёт точку (можно подать как (x,y), так и (Point p)
* Line

Поля:

* + start - точка (Point) начала прямой линии
  + end - точка (Point) конца прямой линии
  + wigth - толщина линии

Методы:

* + Конструктор - задаёт прямую (принимает (xн,yн,xк,yк) или (xн,yн,xк,yк,wigth))
* Cantor

Поля:

* + sizey - толщина элемента множества (прямоугольника)
  + dsizey - расстояние между элементами множества (прямоугольниками)

Методы:

* + Конструктор - задаёт sizey и dsizey
  + set\_float(f) - задаёт dsizey = f
  + count\_len - До какого уровня рекурсии длина элемента множества будет >= 1
  + Draw - инициализация отрисовки фрактала
  + Rec - рекурсивное вычисление и отрисовка элементов фрактала
* Levi

Поля:

Методы:

* + Draw - инициализация отрисовки фрактала
  + Rec - рекурсивное вычисление и отрисовка элементов фрактала
* Gilbert

Поля:

* + sizey - толщина элемента множества (прямоугольника)
  + dsizey - расстояние между элементами множества (прямоугольниками)

Методы:

* + Draw - инициализация отрисовки фрактала
  + Rec - рекурсивное вычисление и отрисовка элементов фрактала
  + Класс P:

Поля:

* + - 5 точек типа Point (p1,p2,p3,p4,p5)
    - w - расстояния между точками
    - direct - направление элемента фрактала

Методы:

* Конструктор - принимает:
  + Координата x начальной точки
  + Координата y начальной точки
  + Расстояние между точками
  + Направление элемента
  + Следующая точка
* Drow - нарисовать линии (p1,p2), (p2,p3), (p3,p4)
  + Класс L:

Поля:

* + - Координаты двух точек (x1,y1) и (x2,y2)
    - dir - направление элемента фрактала

Методы:

* Конструктор – принимает:
  + Координата x начальной точки
  + Координата y начальной точки
  + Координата x конечной точки
  + Координата y конечной точки
  + Направление
  + Класс Lines:

Поля:

* + - Массив линий (элементов класса L)

Методы:

* + - Конструктор без параметров: создаёт пустой массив линий l
* Конструктор: принимает массив line и строит по нему массив l
* ProgressBur

Поля:

* + frac - ссылка на фрактал, для получения информации об отрисовке фрактала
  + isexit - bool переменная, обозначающая нужно ли завершать выполнение
  + timer - таймер для обновления шкалы прогресса отрисовки фрактала
  + progressBar1 - шкала прогресса

Методы:

* + gfrac - Получение ссылки на фрактал
  + Rec - рекурсивное вычисление и отрисовка элементов фрактала