Оглавление

[**Введение** 2](#_Toc106958627)

[**1 Разработка клиентского приложения** 3](#_Toc106958628)

[**1.1 Разработка технического задания** 3](#_Toc106958629)

[*1.1.1 Полное наименование системы и её условное обозначение* 3](#_Toc106958630)

[*1.1.2 Основание для разработки* 3](#_Toc106958631)

[*1.1.3 Назначение и цели создания системы* 3](#_Toc106958632)

[**1.2 Анализ технического задания** 3](#_Toc106958633)

[*1.2.1 Общие положения* 3](#_Toc106958634)

[*1.2.2 Требования к системе* 4](#_Toc106958635)

[**1.3 Стандарты** 5](#_Toc106958636)

[**1.4 Требования к техническим средствам** 5](#_Toc106958637)

[**2 Разработка модели предметной области** 7](#_Toc106958638)

[**2.1 Анализ предметной области** 7](#_Toc106958639)

[**2.2 Разработка структуры классов** 7](#_Toc106958640)

[**3 Разработка структуры приложения** 9](#_Toc106958641)

[**3.1 Разработка архитектуры** 9](#_Toc106958642)

[**3.2 Проектирование пользовательского интерфейса и взаимодействие с ним** 9](#_Toc106958643)

[**Заключение** 15](#_Toc106958644)

[**Список литературы** 16](#_Toc106958645)

[**Приложения** 17](#_Toc106958646)

[**Приложение 1: Код класса Menu:** 17](#_Toc106958647)

[**Приложение 2: Код класса Fractal\_Serpin:** 17](#_Toc106958648)

[**Приложение 3: Код класса Listener:** 32](#_Toc106958649)

[**Приложение 4: Код класса Fractal\_Dragon:** 32](#_Toc106958650)

[**Приложение 5: Код класса Fractal\_Coha:** 37](#_Toc106958651)

**Введение**

В данной пояснительной записке рассматривается описание приложения «Изучение фракталов» на основе объектно-ориентированного подхода.

При объектно-ориентированном подходе программа представляет собой описание объектов, их свойств (или атрибутов), совокупностей (или классов), отношений между ними, способов их взаимодействия и операций над объектами (или методов).

Несомненным преимуществом данного подхода является концептуальная близость к предметной области произвольной структуры и назначения. Механизм наследования атрибутов и методов позволяет строить производные понятия на основе базовых и таким образом создавать модель сколь угодно сложной предметной области с заданными свойствами.

Еще одним теоретически интересным и практически важным свойством объектно-ориентированного подхода является поддержка механизма обработки событий, которые изменяют атрибуты объектов и моделируют их взаимодействие в предметной области.

В отличие от других подходов к программированию, объектно-ориентированный подход требует глубокого понимания основных принципов, или концепций, на которых он базируется.

К числу основополагающих понятий ООП обычно относят абстракцию данных, наследование, инкапсуляцию и полиморфизм.

Объектно-ориентированное программирование в настоящее время является абсолютным лидером в области прикладного программирования.

В качестве основного инструмента разработки применяется Apache NetBeans IDE 12.3. Язык программирования Java.

**1 Разработка клиентского приложения**

**1.1 Разработка технического задания**

*1.1.1 Полное наименование системы и её условное обозначение*

Полное наименование: Разработка приложения для усвоения теоретических знаний «Изучение фракталов» для проведения увлекательно-развивающего досуга.

Условное обозначение системы: «Изучение фракталов».

*1.1.2 Основание для разработки*

Основанием для разработки данной информационной системы является приказ по НГТУ на курсовое проектирование по дисциплине «Программирование на языке Java».

*1.1.3 Назначение и цели создания системы*

1.1.3.1 Назначение системы

Программный продукт предназначен для проведения досуга.

1.1.3.2 Цели создания системы

Программный продукт разрабатывается с целью:

1) увлекательно провести свободное время;

2) развития стратегического мышления и логики.

1.1.3.3 Основные задачи разработки

1) Программа должна предоставлять теоретические знания о построении фракталов и их визуализацию.

2) Пользователь должен интуитивно понимать, как работает данное приложение.

**1.2 Анализ технического задания**

*1.2.1 Общие положения*

Согласно техническому заданию необходимо разработать клиентское приложение, которое предназначено для увлекательного времяпровождения.

*1.2.2 Требования к системе*

1.2.2.1 Требования к структуре и функционированию системы

Программный продукт, разрабатываемый в рамках курсового проекта, должен удовлетворять следующему перечню функциональных требований:

1. Удобный графический интерфейс;
2. Рабочая программа;
3. Наличие визуализации фракталов;

1.2.2.2 Входные данные

Входными данными при работе с программным продуктом является компьютерная мышь.

1.2.2.3 Выходные данные

Выходными данными при работе программы являются данные, выводимые на экран пользователя.

1.2.2.4 Требования к эргономике и технической эстетике

Подсистемы ввода данных, а также формирования и визуализации отчетности должны обеспечивать удобный для конечного пользователя интерфейс.

Меню программного продукта должно позволять пользователю выбирать изучаемый фрактал, а также возможность увидеть информацию о приложении.

Окно приложения «Меню» представляет собой три панели: название приложения, выбор изучаемого фрактала (три варианта) и кнопку для перехода к информации о приложении. Также присутствует три окна с теоретической информацией о выбранном фрактале. На каждом таком окне присутствует четыре составляющие: название выбранного фрактала, кнопка для визуализации фрактала и два информационных элемента. Окно визуализации фрактала для «Треугольника Серпинского» представляет собой пять панелей. Правая панель включает палитру цветов. Левая панель – две кнопки (кнопка «Генерировать» и кнопка «Очистить экран»). Верхняя панель – пояснение к использованию приложения. Центральная – холст; и последняя, нижняя панель, является слайдером, отвечающим за изменение количества итераций. Визуализация остальных фракталов представлена в более минималистичном варианте, в этих окнах присутствует исключительно сама демонстрация фрактала.

**1.3 Стандарты**

Программный продукт разрабатывается на основании следующих государственных стандартов:

1. 2.103-68 ЕСКД. Стадии разработок
2. 2.104-68 ЕСКД. Основные надписи
3. 2.105-95 ЕСКД. Общие требования к текстовым документам
   1. -96 ЕСКД. Текстовые документы
4. 2.111- 68 ЕСКД. Общие требования к текстовым документам.
5. 2.118-73 ЕСКД. Техническое предложение
6. 2.120-73 ЕСКД. Технический проект
7. 2.316-68 ЕСКД. Правила нанесения на чертежах надписей, технических требований и таблиц
8. 7.1-2003 Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления.

**1.4 Требования к техническим средствам**

Для удобства работы система должна обеспечивать отображение GUI с расширенным набором пользовательских элементов, что соответствует платформе Java, NetBeans не ниже версии 12.3, JDK не ниже 16. ОС не важна, главное-виртуальная машина Java выше представленной версии JRE

Минимальные технические характеристики компьютера, на котором гарантируется стабильная работа программы:

1. компьютер/процессор: компьютер-терминал с процессором класса Intel Core2Duo и новее;
2. память: 512 МБ ОЗУ;
3. монитор: монитор с разрешением 800x600 точек или более высоким, поддерживающий 32000 цветов;
4. наличие свободного дискового пространства на жёстком диске - не менее 512 Мб.

**2 Разработка модели предметной области**

**2.1 Анализ предметной области**

Программируемое приложение должно позволять пользователю свободно использовать приложение.

Нефункциональные требования к программному средству:

1. Эффективность: программа должна иметь минимальные требования к аппаратному обеспечению. Не должно требоваться дополнительных периферийных средств (сканер, дополнительные дисководы и т.д.);
2. Понимание: программа должна быть интуитивно понятна, иметь удобный пользовательский интерфейс.

**2.2 Разработка структуры классов**

В соответствие с анализом предметной области можно выделить основные классы информационной системы и их атрибуты:

1. Класс Menu составляет основу для всего приложения
   * JFrame для большинства окон приложения;
   * Переходы между окнами;
2. Класс Fractal\_Serpin содержит основу создания холста и других панелей

* Метод отрисовки внешнего вида окна приложения: пять панель приложения;
* Метод создания слайдера;

1. Класс Listener описывает создание и функционирование элементов для Треугольника Серпинского
   * Ячейки цветов;
   * Слайдера;
   * Активных кнопок;
   * Прослушиватель нажатия мыши;
   * Метод трех вершин;
   * Прорисовка Треугольника Серпинского;
2. Класс Fractal\_Coha описывает создание окна и отрисовку Фрактала Коха;
3. Класс Fractal\_Dragon описывает создание окна и отрисовку Фрактала Дракона.

**3 Разработка структуры приложения**

**3.1 Разработка архитектуры**

При разработке приложения мы будем реализовывать 2 механизма:

* Окно меню и открытие остальных окон – 1 класс;
* Функциональное окно – 1 класс;
* Работа с панелями – 1 класс;
* Визуализация фракталов – 2 класса.

**3.2 Проектирование пользовательского интерфейса и взаимодействие с ним**

В данном проекте для продуктивной работы реализован удобный пользовательский интерфейс, похожий на большинство Windows – приложений.

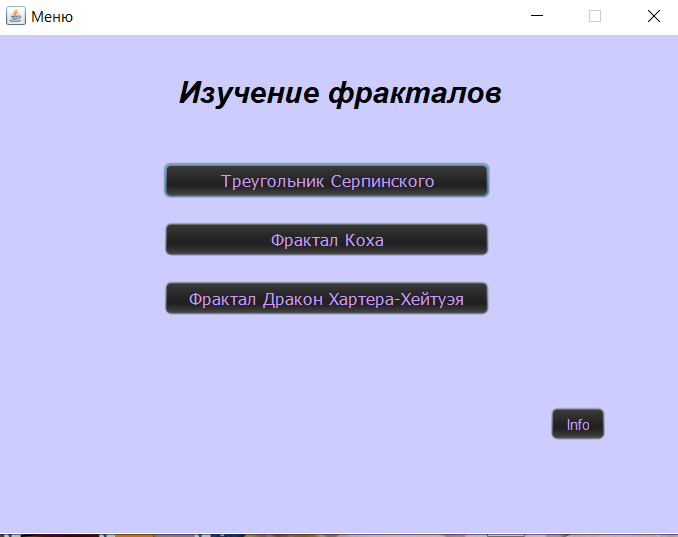
Начальное окно приложения является меню данного приложения (рис.1). С него можно открыть информационные окна с описанием фракталов, с которых уже можно открыть визуализацию данных элементов. 

Рис. 1 – Интерфейс приложения. Окно меню.

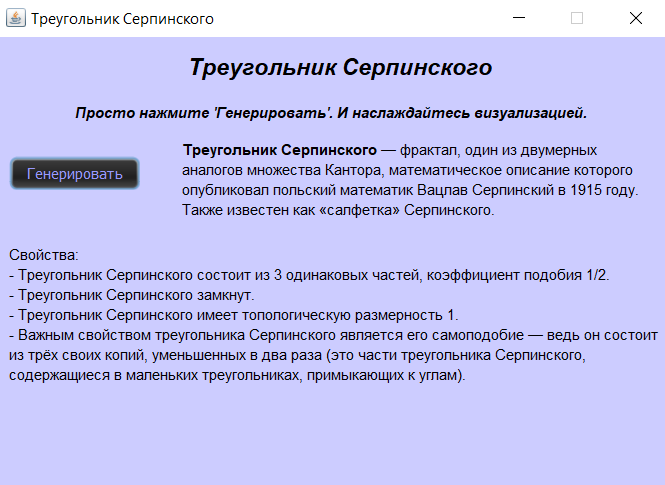


Рис. 2 – Информационное окно для Треугольника Серпинского.

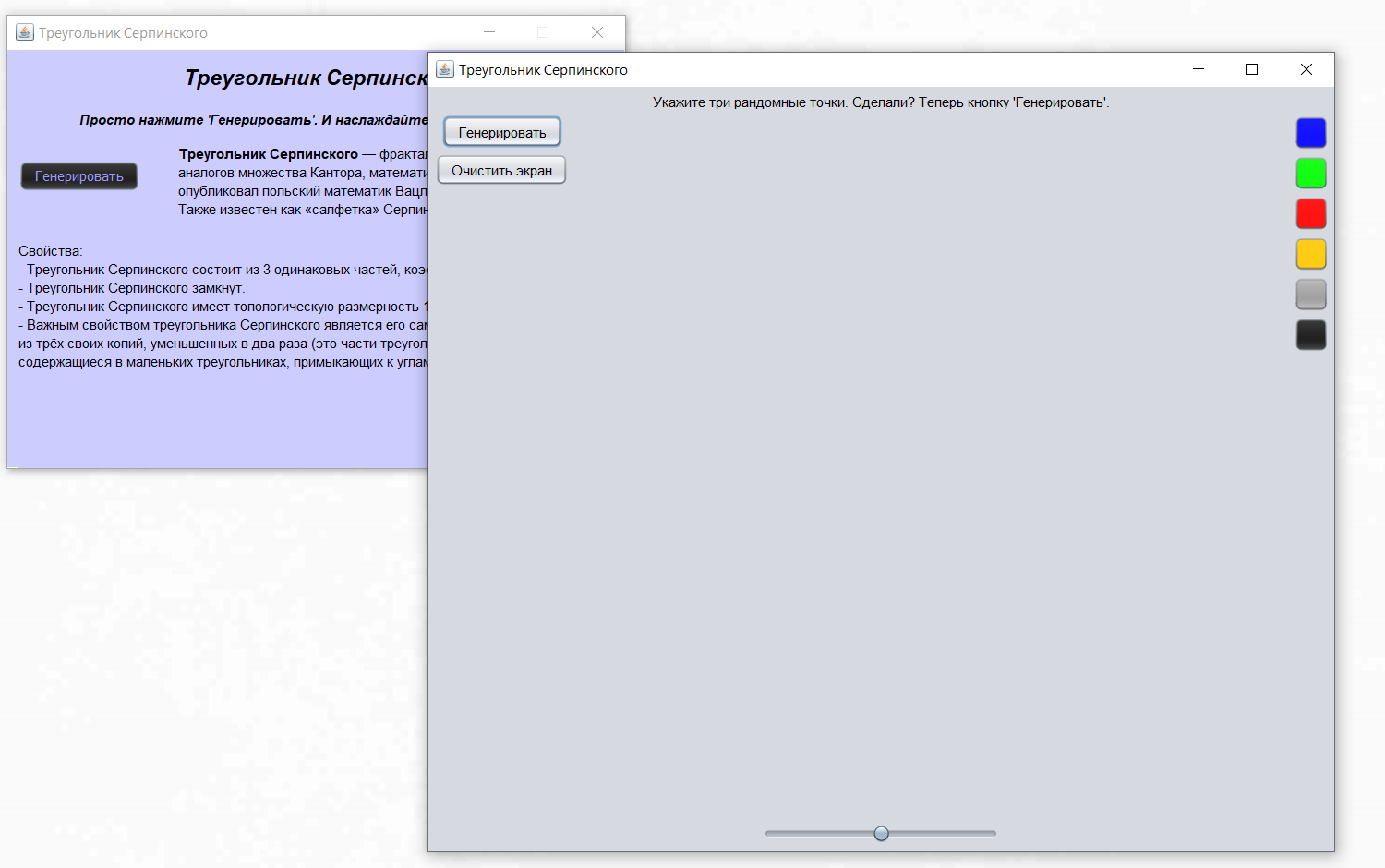


Рис. 3 – Интерфейс окна «Треугольник Серпинского»

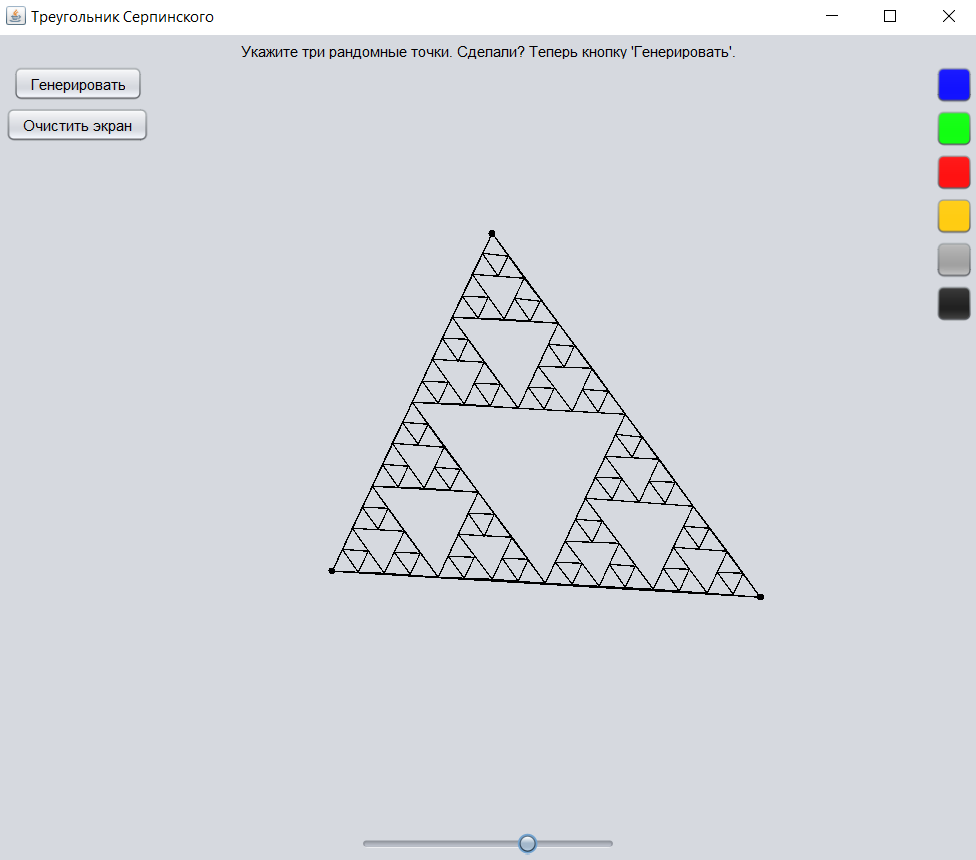


Рис. 4 – Интерфейс окна во время использования.



Рис. 5 – Правая панель приложения: изменение цвета.

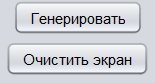


Рис. 6 – Левая панель: кнопки.



Рис. 7 – Нижняя панель: слайдер.

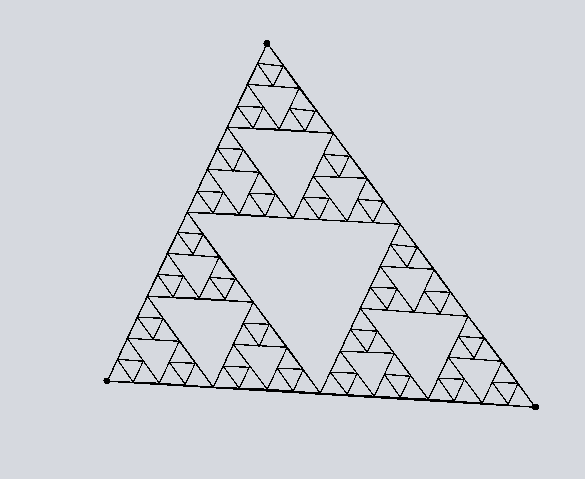


Рис. 8 – Центральная панель: холст.

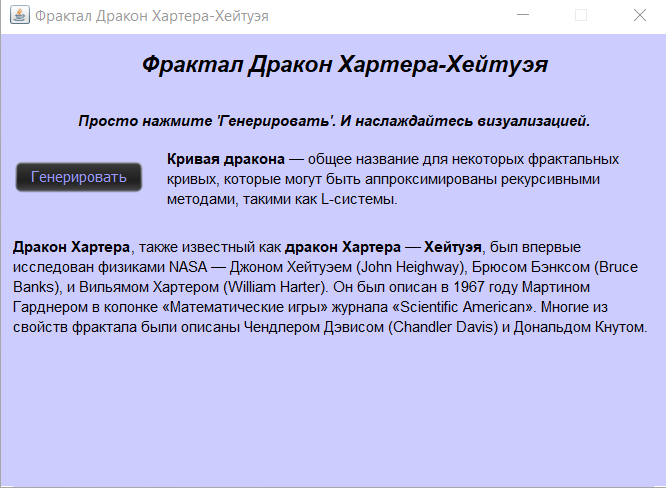


Рис. 9 – Информационное окно для Фрактала Дракона Хартера – Хейтуэя.

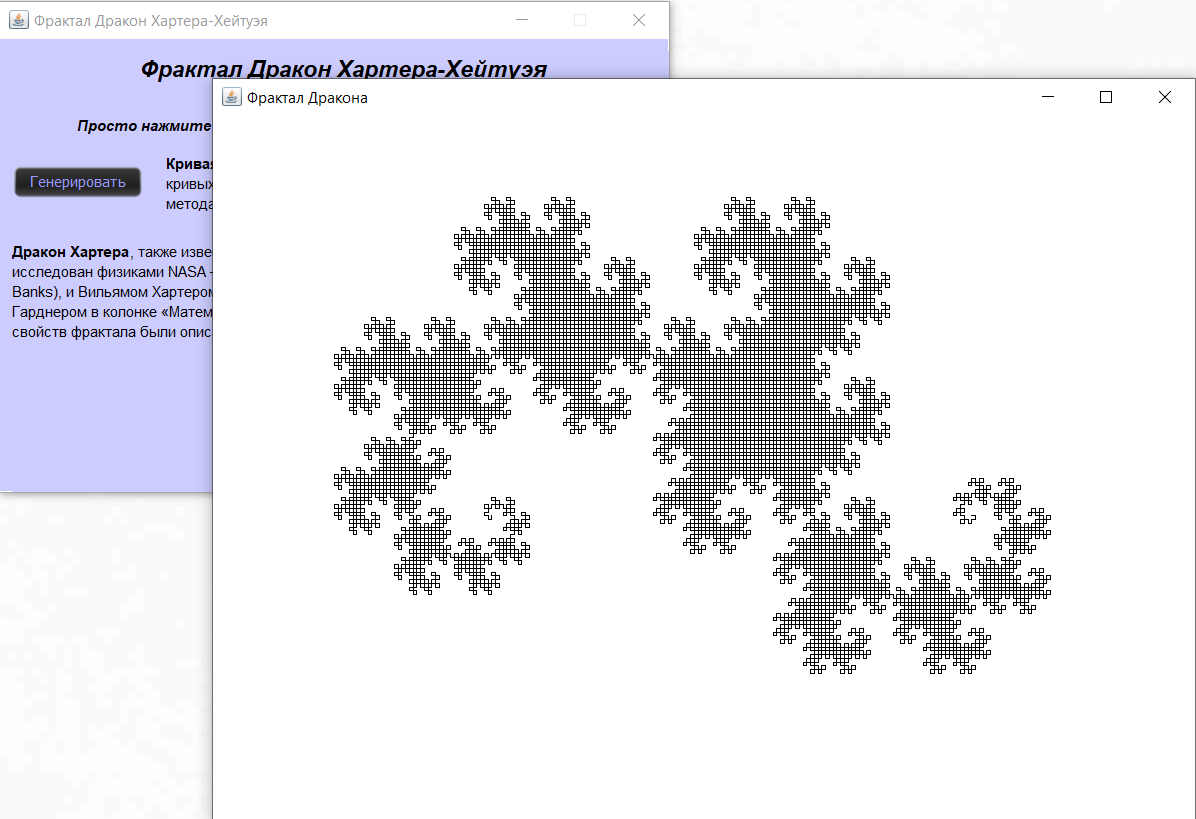


Рис. 10 – Интерфейс окна «Фрактал Дракона».

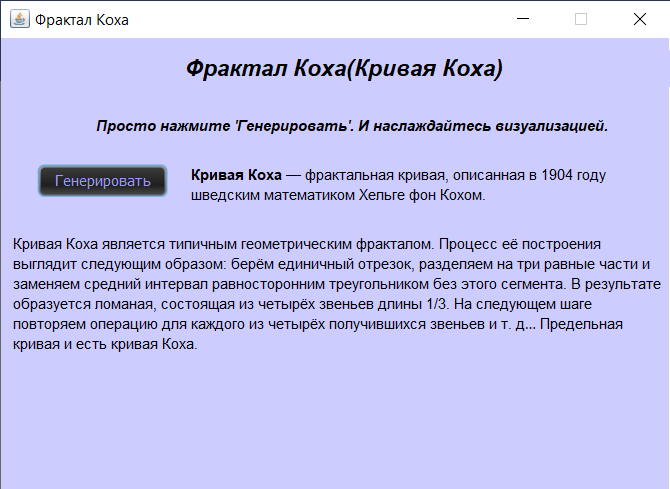


Рис. 11 – Информационное окно для Фрактала Коха.

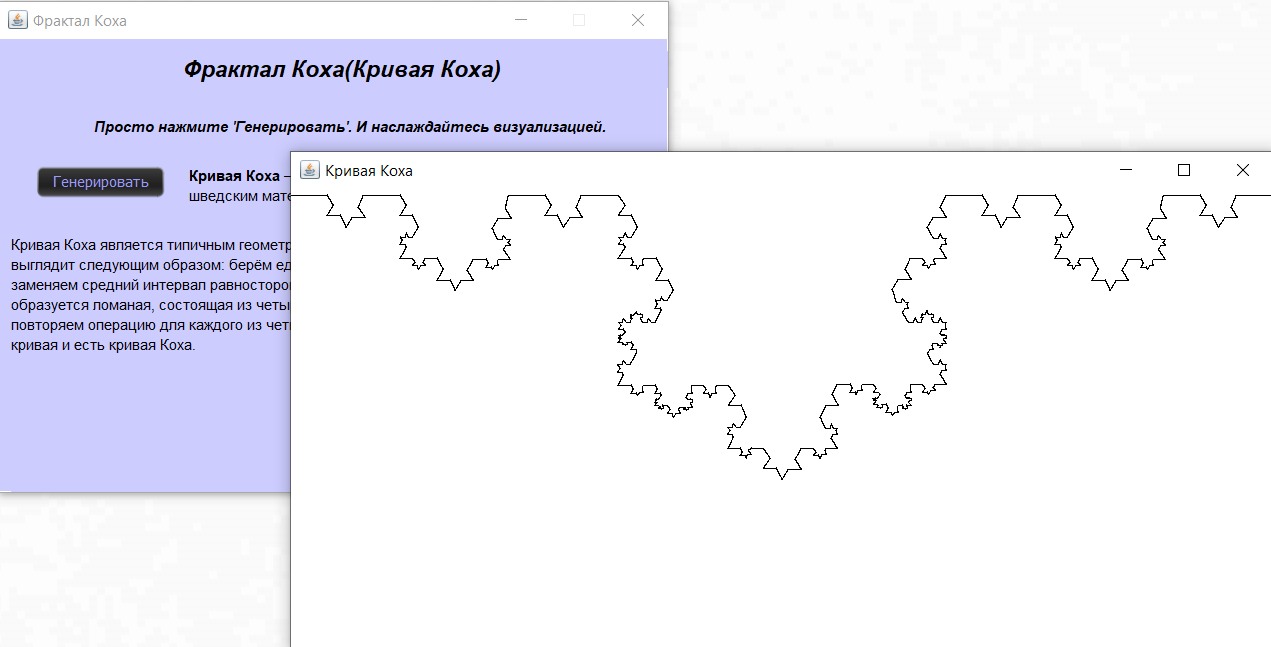


Рис. 12 – Интерфейс окна «Фрактал Коха».

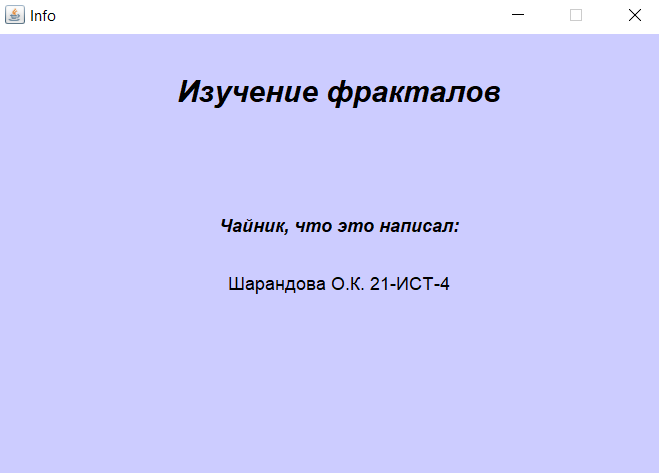


Рис. 13 – Окно информации.

Основной сценарий работы пользователя с приложением:

1. При запуске приложения открывается навигационное меню.
2. При выборе фрактала «Треугольник Серпинского» открывается информационное окно.
3. При нажатии на кнопку «Генерировать» запускается окно для изучения «Треугольник Серпинского».
4. При запуске окна запускается чистый холст.
5. Пользователь видит текстовое пояснение: нужно создать три точки.
6. Пользователь, используя мышку выставляет точки.
7. Пользователь может создать треугольник на основе трех точек с помощью кнопки «Генерировать» или начать заново с помощью кнопки «Очистить экран».
8. Пользователь может изменить цвет по своему усмотрению.
9. При выборе остальных фракталов также открываются информационные окна.
10. При нажатии на кнопку «Генерировать» запускаются окна для изучения фракталов.

**Заключение**

В данной курсовой работе была разработано приложение «Изучение фракталов» на основе объектно-ориентированного подхода. Этот метод позволяет обеспечить разделение данных, управление доступом к ним, а также расширяемость и модифицируемость.

В проекте удалось реализовать все поставленные задачи:

1. Создать удобный графический интерфейс;
2. Спроектировать рабочую программа;
3. Создать возможность изучения фракталов с теоретической, а также визуальной составляющей.

Однако, остается широкое поле для дальнейшей деятельности по развитию и улучшению проекта.

Таким образом, следует продолжить работу над проектом в будущем и выполнить определенные в ходе анализа проведенной деятельности задачи по доработке программы.

**Список литературы**

1. <https://stackoverflow.com>
2. <https://www.youtube.com>
3. Маслов В.В Основы программирования на Java
4. Арнолд Кен Язык программирования Java
5. Мейнджер Д. Java: основы программирования
6. Шилдт Г Java. Методики программирования

**Приложения**

**Приложение 1: Код класса Menu:**

package fractals1;

public class Menu extends javax.swing.JFrame {

/\*\*

\* Creates new form Menu

\*/

public Menu() {

initComponents();

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jFrame1 = new javax.swing.JFrame();

jPanel2 = new javax.swing.JPanel();

label2 = new java.awt.Label();

label3 = new java.awt.Label();

jButton6 = new javax.swing.JButton();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jFrame2 = new javax.swing.JFrame();

jPanel3 = new javax.swing.JPanel();

label4 = new java.awt.Label();

label5 = new java.awt.Label();

jButton9 = new javax.swing.JButton();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jFrame3 = new javax.swing.JFrame();

jPanel4 = new javax.swing.JPanel();

label6 = new java.awt.Label();

label7 = new java.awt.Label();

jButton12 = new javax.swing.JButton();

jLabel5 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

jFrame5 = new javax.swing.JFrame();

jPanel6 = new javax.swing.JPanel();

label10 = new java.awt.Label();

label11 = new java.awt.Label();

label12 = new java.awt.Label();

jPanel1 = new javax.swing.JPanel();

label1 = new java.awt.Label();

jButton4 = new javax.swing.JButton();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jButton5 = new javax.swing.JButton();

jFrame1.setTitle("Треугольник Серпинского");

jFrame1.setMaximumSize(new java.awt.Dimension(550, 400));

jFrame1.setMinimumSize(new java.awt.Dimension(550, 400));

jFrame1.setPreferredSize(new java.awt.Dimension(550, 400));

jFrame1.setResizable(false);

jPanel2.setBackground(new java.awt.Color(204, 204, 255));

jPanel2.setMaximumSize(new java.awt.Dimension(550, 400));

jPanel2.setMinimumSize(new java.awt.Dimension(550, 400));

jPanel2.setPreferredSize(new java.awt.Dimension(550, 400));

label2.setAlignment(java.awt.Label.CENTER);

label2.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT\_CURSOR));

label2.setFont(new java.awt.Font("Segoe UI", 3, 18)); // NOI18N

label2.setText("Треугольник Серпинского");

label3.setAlignment(java.awt.Label.CENTER);

label3.setFont(new java.awt.Font("Dialog", 3, 12)); // NOI18N

label3.setMinimumSize(new java.awt.Dimension(400, 15));

label3.setText("Просто нажмите 'Генерировать'. И наслаждайтесь визуализацией.");

jButton6.setBackground(new java.awt.Color(0, 0, 0));

jButton6.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

jButton6.setForeground(new java.awt.Color(153, 153, 255));

jButton6.setText("Генерировать");

jButton6.setToolTipText("");

jButton6.addActionListener(evt -> new Fractal\_Serpin());

jLabel3.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

jLabel3.setText("<html><b>Треугольник Серпинского</b> — фрактал, один из двумерных аналогов множества Кантора, математическое описание которого опубликовал польский математик Вацлав Серпинский в 1915 году. Также известен как «салфетка» Серпинского.</html>");

jLabel3.setVerticalAlignment(javax.swing.SwingConstants.TOP);

jLabel4.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

jLabel4.setText("<html>Свойства:<br>- Треугольник Серпинского состоит из 3 одинаковых частей, коэффициент подобия 1/2.<br>- Треугольник Серпинского замкнут.<br>- Треугольник Серпинского имеет топологическую размерность 1.<br>- Важным свойством треугольника Серпинского является его самоподобие — ведь он состоит из трёх своих копий, уменьшенных в два раза (это части треугольника Серпинского, содержащиеся в маленьких треугольниках, примыкающих к углам).</html>");

jLabel4.setVerticalAlignment(javax.swing.SwingConstants.TOP);

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);

jPanel2.setLayout(jPanel2Layout);

jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addGap(10, 10, 10)

.addComponent(label3, javax.swing.GroupLayout.PREFERRED\_SIZE, 495, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(35, Short.MAX\_VALUE))

.addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(label2, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout.createSequentialGroup()

.addComponent(jButton6)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED\_SIZE, 379, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(13, 13, 13))

.addComponent(jLabel4, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addContainerGap())))

);

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addContainerGap()

.addComponent(label2, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(label3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED\_SIZE, 66, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel2Layout.createSequentialGroup()

.addGap(24, 24, 24)

.addComponent(jButton6)))

.addGap(18, 18, 18)

.addComponent(jLabel4, javax.swing.GroupLayout.DEFAULT\_SIZE, 227, Short.MAX\_VALUE)

.addContainerGap())

);

javax.swing.GroupLayout jFrame1Layout = new javax.swing.GroupLayout(jFrame1.getContentPane());

jFrame1.getContentPane().setLayout(jFrame1Layout);

jFrame1Layout.setHorizontalGroup(

jFrame1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel2, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

);

jFrame1Layout.setVerticalGroup(

jFrame1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel2, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

);

jFrame2.setTitle("Фрактал Коха");

jFrame2.setMaximumSize(new java.awt.Dimension(550, 400));

jFrame2.setMinimumSize(new java.awt.Dimension(550, 400));

jFrame2.setPreferredSize(new java.awt.Dimension(550, 400));

jFrame2.setResizable(false);

jPanel3.setBackground(new java.awt.Color(204, 204, 255));

jPanel3.setMaximumSize(new java.awt.Dimension(550, 400));

jPanel3.setMinimumSize(new java.awt.Dimension(550, 400));

jPanel3.setPreferredSize(new java.awt.Dimension(550, 400));

label4.setAlignment(java.awt.Label.CENTER);

label4.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT\_CURSOR));

label4.setFont(new java.awt.Font("Segoe UI", 3, 18)); // NOI18N

label4.setText("Фрактал Коха(Кривая Коха)");

label5.setAlignment(java.awt.Label.CENTER);

label5.setFont(new java.awt.Font("Dialog", 3, 12)); // NOI18N

label5.setMinimumSize(new java.awt.Dimension(400, 15));

label5.setText("Просто нажмите 'Генерировать'. И наслаждайтесь визуализацией.");

jButton9.setBackground(new java.awt.Color(0, 0, 0));

jButton9.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

jButton9.setForeground(new java.awt.Color(153, 153, 255));

jButton9.setText("Генерировать");

jButton9.setToolTipText("");

jButton9.addActionListener(evt -> new Fractal\_Coha());

jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

jLabel1.setText("<html><b>Кривая Коха</b> — фрактальная кривая, описанная в 1904 году шведским математиком Хельге фон Кохом.</html>");

jLabel1.setVerticalAlignment(javax.swing.SwingConstants.TOP);

jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

jLabel2.setText("<html>Кривая Коха является типичным геометрическим фракталом. Процесс её построения выглядит следующим образом: берём единичный отрезок, разделяем на три равные части и заменяем средний интервал равносторонним треугольником без этого сегмента. В результате образуется ломаная, состоящая из четырёх звеньев длины 1/3. На следующем шаге повторяем операцию для каждого из четырёх получившихся звеньев и т. д… Предельная кривая и есть кривая Коха.</html>");

jLabel2.setVerticalAlignment(javax.swing.SwingConstants.TOP);

javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);

jPanel3.setLayout(jPanel3Layout);

jPanel3Layout.setHorizontalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(label4, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel3Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel3Layout.createSequentialGroup()

.addComponent(label5, javax.swing.GroupLayout.PREFERRED\_SIZE, 470, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(33, 33, 33))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel3Layout.createSequentialGroup()

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT\_SIZE, 522, Short.MAX\_VALUE)

.addGroup(jPanel3Layout.createSequentialGroup()

.addComponent(jButton9)

.addGap(18, 18, 18)

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 380, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGap(18, 18, 18))))

);

jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel3Layout.createSequentialGroup()

.addContainerGap()

.addComponent(label4, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(label5, javax.swing.GroupLayout.PREFERRED\_SIZE, 41, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 49, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton9))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT\_SIZE, 235, Short.MAX\_VALUE)

.addContainerGap())

);

javax.swing.GroupLayout jFrame2Layout = new javax.swing.GroupLayout(jFrame2.getContentPane());

jFrame2.getContentPane().setLayout(jFrame2Layout);

jFrame2Layout.setHorizontalGroup(

jFrame2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel3, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

jFrame2Layout.setVerticalGroup(

jFrame2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel3, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

);

jFrame3.setTitle("Фрактал Дракон Хартера-Хейтуэя");

jFrame3.setMaximumSize(new java.awt.Dimension(550, 400));

jFrame3.setMinimumSize(new java.awt.Dimension(550, 400));

jFrame3.setPreferredSize(new java.awt.Dimension(550, 400));

jFrame3.setResizable(false);

jPanel4.setBackground(new java.awt.Color(204, 204, 255));

jPanel4.setMaximumSize(new java.awt.Dimension(550, 400));

jPanel4.setMinimumSize(new java.awt.Dimension(550, 400));

jPanel4.setName(""); // NOI18N

jPanel4.setPreferredSize(new java.awt.Dimension(550, 400));

label6.setAlignment(java.awt.Label.CENTER);

label6.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT\_CURSOR));

label6.setFont(new java.awt.Font("Segoe UI", 3, 18)); // NOI18N

label6.setText("Фрактал Дракон Хартера-Хейтуэя");

label7.setAlignment(java.awt.Label.CENTER);

label7.setFont(new java.awt.Font("Dialog", 3, 12)); // NOI18N

label7.setMinimumSize(new java.awt.Dimension(400, 15));

label7.setText("Просто нажмите 'Генерировать'. И наслаждайтесь визуализацией.");

jButton12.setBackground(new java.awt.Color(0, 0, 0));

jButton12.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

jButton12.setForeground(new java.awt.Color(153, 153, 255));

jButton12.setText("Генерировать");

jButton12.setToolTipText("");

jButton12.addActionListener(evt -> new Fractal\_Dragon(14));

jLabel5.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

jLabel5.setText("<html><b>Кривая дракона</b> — общее название для некоторых фрактальных кривых, которые могут быть аппроксимированы рекурсивными методами, такими как L-системы.</html>");

jLabel5.setVerticalAlignment(javax.swing.SwingConstants.TOP);

jLabel6.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

jLabel6.setText("<html><b>Дракон Хартера</b>, также известный как <b>дракон Хартера — Хейтуэя</b>, был впервые исследован физиками NASA — Джоном Хейтуэем (John Heighway), Брюсом Бэнксом (Bruce Banks), и Вильямом Хартером (William Harter). Он был описан в 1967 году Мартином Гарднером в колонке «Математические игры» журнала «Scientific American». Многие из свойств фрактала были описаны Чендлером Дэвисом (Chandler Davis) и Дональдом Кнутом.</html>");

jLabel6.setVerticalAlignment(javax.swing.SwingConstants.TOP);

javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);

jPanel4.setLayout(jPanel4Layout);

jPanel4Layout.setHorizontalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(label6, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(jPanel4Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel4Layout.createSequentialGroup()

.addComponent(jButton12)

.addGap(18, 18, 18)

.addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED\_SIZE, 386, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(label7, javax.swing.GroupLayout.PREFERRED\_SIZE, 514, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED\_SIZE, 513, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(26, Short.MAX\_VALUE))

);

jPanel4Layout.setVerticalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel4Layout.createSequentialGroup()

.addContainerGap()

.addComponent(label6, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(19, 19, 19)

.addComponent(label7, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel4Layout.createSequentialGroup()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED\_SIZE, 59, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel4Layout.createSequentialGroup()

.addGap(20, 20, 20)

.addComponent(jButton12)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jLabel6, javax.swing.GroupLayout.DEFAULT\_SIZE, 233, Short.MAX\_VALUE)

.addContainerGap())

);

javax.swing.GroupLayout jFrame3Layout = new javax.swing.GroupLayout(jFrame3.getContentPane());

jFrame3.getContentPane().setLayout(jFrame3Layout);

jFrame3Layout.setHorizontalGroup(

jFrame3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel4, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

);

jFrame3Layout.setVerticalGroup(

jFrame3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel4, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

);

jFrame5.setTitle("Info");

jFrame5.setMaximumSize(new java.awt.Dimension(550, 400));

jFrame5.setMinimumSize(new java.awt.Dimension(550, 400));

jFrame5.setPreferredSize(new java.awt.Dimension(550, 400));

jFrame5.setResizable(false);

jPanel6.setBackground(new java.awt.Color(204, 204, 255));

jPanel6.setMaximumSize(new java.awt.Dimension(550, 400));

jPanel6.setMinimumSize(new java.awt.Dimension(550, 400));

jPanel6.setPreferredSize(new java.awt.Dimension(546, 360));

label10.setAlignment(java.awt.Label.CENTER);

label10.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT\_CURSOR));

label10.setFont(new java.awt.Font("Segoe UI", 3, 24)); // NOI18N

label10.setText("Изучение фракталов");

label11.setAlignment(java.awt.Label.CENTER);

label11.setFont(new java.awt.Font("Dialog", 3, 14)); // NOI18N

label11.setText("Чайник, что это написал:");

label12.setAlignment(java.awt.Label.CENTER);

label12.setFont(new java.awt.Font("Dialog", 0, 14)); // NOI18N

label12.setText("Шарандова О.К. 21-ИСТ-4");

javax.swing.GroupLayout jPanel6Layout = new javax.swing.GroupLayout(jPanel6);

jPanel6.setLayout(jPanel6Layout);

jPanel6Layout.setHorizontalGroup(

jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel6Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(label11, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(label12, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(label10, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, 530, Short.MAX\_VALUE))

.addContainerGap())

);

jPanel6Layout.setVerticalGroup(

jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel6Layout.createSequentialGroup()

.addContainerGap()

.addComponent(label10, javax.swing.GroupLayout.PREFERRED\_SIZE, 74, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(57, 57, 57)

.addComponent(label11, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(19, 19, 19)

.addComponent(label12, javax.swing.GroupLayout.PREFERRED\_SIZE, 29, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(188, Short.MAX\_VALUE))

);

javax.swing.GroupLayout jFrame5Layout = new javax.swing.GroupLayout(jFrame5.getContentPane());

jFrame5.getContentPane().setLayout(jFrame5Layout);

jFrame5Layout.setHorizontalGroup(

jFrame5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel6, javax.swing.GroupLayout.Alignment.TRAILING, 550, 550, javax.swing.GroupLayout.PREFERRED\_SIZE)

);

jFrame5Layout.setVerticalGroup(

jFrame5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel6, 400, 400, javax.swing.GroupLayout.PREFERRED\_SIZE)

);

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Меню");

setMaximumSize(new java.awt.Dimension(550, 400));

setMinimumSize(new java.awt.Dimension(550, 400));

setResizable(false);

jPanel1.setBackground(new java.awt.Color(204, 204, 255));

jPanel1.setMaximumSize(new java.awt.Dimension(550, 400));

jPanel1.setMinimumSize(new java.awt.Dimension(550, 400));

jPanel1.setPreferredSize(new java.awt.Dimension(550, 400));

label1.setAlignment(java.awt.Label.CENTER);

label1.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT\_CURSOR));

label1.setFont(new java.awt.Font("Segoe UI", 3, 24)); // NOI18N

label1.setText("Изучение фракталов");

jButton4.setBackground(new java.awt.Color(0, 0, 0));

jButton4.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

jButton4.setForeground(new java.awt.Color(204, 153, 255));

jButton4.setText("Треугольник Серпинского");

jButton4.addActionListener(evt -> jFrame1.show());

jButton1.setBackground(new java.awt.Color(0, 0, 0));

jButton1.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

jButton1.setForeground(new java.awt.Color(204, 153, 255));

jButton1.setText("Фрактал Коха");

jButton1.setMaximumSize(new java.awt.Dimension(253, 25));

jButton1.setMinimumSize(new java.awt.Dimension(253, 25));

jButton1.addActionListener(evt -> jFrame2.show());

jButton2.setBackground(new java.awt.Color(0, 0, 0));

jButton2.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

jButton2.setForeground(new java.awt.Color(204, 153, 255));

jButton2.setText("Фрактал Дракон Хартера-Хейтуэя");

jButton2.addActionListener(evt -> jFrame3.show());

jButton5.setBackground(new java.awt.Color(0, 0, 0));

jButton5.setForeground(new java.awt.Color(204, 153, 255));

jButton5.setText("Info");

jButton5.addActionListener(evt -> jFrame5.show());

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addComponent(label1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(133, 133, 133)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jButton1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton4, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton2, javax.swing.GroupLayout.DEFAULT\_SIZE, 262, Short.MAX\_VALUE))

.addGap(0, 145, Short.MAX\_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton5)

.addGap(52, 52, 52)))

.addContainerGap())

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addComponent(label1, javax.swing.GroupLayout.PREFERRED\_SIZE, 74, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(jButton4)

.addGap(18, 18, 18)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(jButton2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 91, Short.MAX\_VALUE)

.addComponent(jButton5)

.addGap(75, 75, 75))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

pack();

}// </editor-fold>

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Menu.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Menu.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Menu.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Menu.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new Menu().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton12;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton4;

private javax.swing.JButton jButton5;

private javax.swing.JButton jButton6;

private javax.swing.JButton jButton9;

private javax.swing.JFrame jFrame1;

private javax.swing.JFrame jFrame2;

private javax.swing.JFrame jFrame3;

private javax.swing.JFrame jFrame5;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JPanel jPanel3;

private javax.swing.JPanel jPanel4;

private javax.swing.JPanel jPanel6;

private java.awt.Label label1;

private java.awt.Label label10;

private java.awt.Label label11;

private java.awt.Label label12;

private java.awt.Label label2;

private java.awt.Label label3;

private java.awt.Label label4;

private java.awt.Label label5;

private java.awt.Label label6;

private java.awt.Label label7;

// End of variables declaration

}

**Приложение 2: Код класса Fractal\_Serpin:**

package fractals1;

import java.awt.BorderLayout;

import java.awt.Color;

import java.awt.Dimension;

import java.awt.FlowLayout;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JSlider;

public class Fractal\_Serpin extends JFrame {

public Fractal\_Serpin() {

init();

}

private void init() {

// Создать окно для рисования

JFrame jf = new JFrame("Треугольник Серпинского");

// Создать объект слушателя

Listener ls = new Listener();

// Установить свойства панели

jf.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

jf.setSize(800, 700); // Установить размер 800 \* 700

jf.setLocationRelativeTo(null);// Установить отображение по центру

jf.setLayout(new BorderLayout());// Установить макет границы (JFrame по умолчанию использует макет границы)

// Создать панель (а Panel представляет собой потоковый макет)

JPanel jp1 = new JPanel();

JPanel jp2 = new JPanel();

JPanel jp3 = new JPanel();

JPanel jp4 = new JPanel();

JPanel jp5 = new JPanel();

jp1.setLayout(new FlowLayout());

jp2.setLayout(new FlowLayout());

jp1.setPreferredSize(new Dimension(130, 100)); //Для кнопок

jp2.setPreferredSize(new Dimension(40, 100)); //Для иконок цвета

jp5.setPreferredSize(new Dimension(100, 20)); //Для задания

jf.add(jp1, BorderLayout.WEST); //кнопки

jf.add(jp2, BorderLayout.EAST); //цвета

jf.add(jp3, BorderLayout.CENTER); //область для треугольника

jf.add(jp4, BorderLayout.SOUTH); //слайдер

jf.add(jp5, BorderLayout.NORTH); //задание

jp3.addMouseMotionListener(ls);

jp3.addMouseListener(ls);

// Создать слайдер

JSlider JS = new JSlider();

// Добавить слайдер

jp4.add(JS);

JS.setValue(3);

JS.setSnapToTicks(true);

JS.setMaximum(6);

JS.setMinimum(0);

JS.addChangeListener(ls);

//Создание области для пояснения

JLabel JL = new JLabel("Укажите три рандомные точки. Сделали? Теперь кнопку 'Генерировать'.");

//Добавить пояснение

jp5.add(JL);

// Создание кнопок с массивами и циклами и добавление слушателей. Формат: type [] array name = {"variable", "variable" ...};

String[] ShapeBtn = {"Генерировать", "Очистить экран"};

for (String ShapeBtn1 : ShapeBtn) {

JButton shape = new JButton(ShapeBtn1);

jp1.add(shape);

shape.addActionListener(ls);

}

// Создать цветную кнопку

Color[] ColorBtn = {Color.BLUE, Color.GREEN, Color.RED, Color.ORANGE, Color.GRAY, Color.BLACK};

Dimension colorBtnSize = new Dimension(30, 30); // Устанавливаем размер цветного квадрата

for (Color ColorBtn1 : ColorBtn) {

JButton color = new JButton(); // Создать цвет кнопки объекта

color.setBackground(ColorBtn1); // Установить цвет фона кнопки

color.setPreferredSize(colorBtnSize);

jp2.add(color);

color.addActionListener(ls);

}

// Установить окно видимым

jf.setVisible(true);

// Получить холст формы и передать его объекту слушателя (должен быть виден после установки окна)

ls.g = jp3.getGraphics();

}

}

**Приложение 3: Код класса Listener:**

package fractals1;

import java.awt.Color;

import java.awt.Graphics;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import java.awt.event.MouseMotionListener;

import javax.swing.JButton;

import javax.swing.JSlider;

import javax.swing.event.ChangeEvent;

import javax.swing.event.ChangeListener;

public class Listener implements ActionListener, MouseListener, MouseMotionListener, ChangeListener {

int x1;

int y1;

int x2;

int y2;

int x3;

int y3;

String action;

Graphics g;

int[] X = new int[3];

int[] Y = new int[3];

int index = 0;

int n = 3;

public void x(int x1, int y1, int x2, int y2, int x3, int y3, int n) {

index = 0;

if (n > 0) {

n--;

g.drawLine(x1, y1, x2, y2);

g.drawLine(x3, y3, x2, y2);

g.drawLine(x1, y1, x3, y3);

int x4 = (x1 + x3) / 2;

int y4 = (y1 + y3) / 2;

int x5 = (x1 + x2) / 2;

int y5 = (y2 + y1) / 2;

int x6 = (x3 + x2) / 2;

int y6 = (y3 + y2) / 2;

g.drawLine(x4, y4, x5, y5);

g.drawLine(x6, y6, x5, y5);

g.drawLine(x4, y4, x6, y6);

// Итерация

x(x1, y1, x5, y5, x4, y4, n);

x(x5, y5, x2, y2, x6, y6, n);

x(x4, y4, x6, y6, x3, y3, n);

}

}

/\*\*

\*

\*/

@Override

public void mouseClicked(MouseEvent e) {

if (index < 3) {

// Получить координаты трех вершин

X[index] = e.getX();

Y[index] = e.getY();

System.out.println("Нет." + (index + 1) + "Точка");

g.fillOval(X[index] - 3, Y[index] - 3, 6, 6);

index++;

}

// Сохраняем координаты в x\_i, y\_i

x1 = X[0];

x2 = X[1];

x3 = X[2];

y1 = Y[0];

y2 = Y[1];

y3 = Y[2];

}

public void actionPerformed(ActionEvent e) {

action = e.getActionCommand();

if (action.equals("")) { // цвет

// Получить источник события

Object srcObj = e.getSource();

// Получить кнопку

JButton srcBtn = (JButton) srcObj;

// Получить цвет

Color color = srcBtn.getBackground();

g.setColor(color);

} else if ("Очистить экран".equals(action)) {

g.clearRect(0, 0, 2000, 2000);

} else if (action.equals("Генерировать")) {

x(x1, y1, x2, y2, x3, y3, n);

}

}

@Override

public void stateChanged(ChangeEvent e) {

// Получить объект слайдера

Object Ob = e.getSource();

JSlider Obj = (JSlider) Ob;

n = Obj.getValue();

// Очистить экран

g.clearRect(0, 0, 2000, 2000);

// нарисовать вершину

for (int i = 0; i < 3; i++) {

g.fillOval(X[i] - 3, Y[i] - 3, 6, 6);

}

// Рекурсивный

x(x1, y1, x2, y2, x3, y3, n);

}

@Override

public void mousePressed(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mouseReleased(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mouseEntered(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mouseExited(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mouseDragged(MouseEvent e) {

// TODO Auto-generated method stub

}

/\*\*

\*

\*/

@Override

public void mouseMoved(MouseEvent e) {

// TODO Auto-generated method stub

}

}

**Приложение 4: Код класса Fractal\_Dragon:**

package fractals1;

import java.awt.Color;

import java.awt.Graphics;

import java.util.ArrayList;

import java.util.Collections;

import java.util.List;

import javax.swing.JFrame;

public final class Fractal\_Dragon extends JFrame {

public List<Integer> turns;

public double startingAngle;

public double side;

public Fractal\_Dragon(int iter) {

super("Фрактал Дракона");

setBounds(100, 100, 800, 600);

setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);

turns = getSequence(iter);

startingAngle = -iter \* (Math.PI / 4);

side = 400 / Math.pow(2, iter / 2.);

setVisible(true);

getSequence(PROPERTIES);

}

public List<Integer> getSequence(int iterations) {

List<Integer> turnSequence = new ArrayList<>();

for (int i = 0; i < iterations; i++) {

List<Integer> copy = new ArrayList<>(turnSequence);

Collections.reverse(copy);

turnSequence.add(1);

copy.forEach(turn -> {

turnSequence.add(-turn);

});

}

return turnSequence;

}

@Override

public void paint(Graphics g) {

g.setColor(Color.BLACK);

double angle = startingAngle;

int x1 = 230, y1 = 350;

int x2 = x1 + (int) (Math.cos(angle) \* side);

int y2 = y1 + (int) (Math.sin(angle) \* side);

g.drawLine(x1, y1, x2, y2);

x1 = x2;

y1 = y2;

for (Integer turn : turns) {

angle += turn \* (Math.PI / 2);

x2 = x1 + (int) (Math.cos(angle) \* side);

y2 = y1 + (int) (Math.sin(angle) \* side);

g.drawLine(x1, y1, x2, y2);

x1 = x2;

y1 = y2;

}

}

}

**Приложение 5: Код класса Fractal\_Coha:**

package fractals1;

import java.awt.Canvas;

import java.awt.Color;

import java.awt.Frame;

import java.awt.Graphics;

import java.awt.Point;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

import javax.swing.JFrame;

public class Fractal\_Coha extends JFrame {

public static int width = 800, height = 600;

/\*\*

\* Создаёт, настраивает и выводит фрейм для прорисовки кривой Коха.

\*/

public Fractal\_Coha() {

super("Кривая Коха");

setSize(width, height);

setVisible(true);

setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);

}

/\*\*

\* Формирует цвета и масштаб, рекурсивно вызывает метод

\*/

public void paint(Graphics g) {

setBackground(Color.WHITE);

g.setColor(Color.BLACK);

Point a = new Point(0, 6);

Point b = new Point(this.getWidth(), 6);

drawKochLine(g, a, b, 0, 6);

}

/\*\*

\* @param g the specified Graphics context

\* @param a начальная точка линии

\* @param b конечная точка линии

\* @param fi угол поворота линии

\* @param n оставшаяся глубина рекурсии

\*/

public void drawKochLine(Graphics g, Point a, Point b, double fi, int n) {

if (n <= 0) {

// рисуем прямую, если достигнута необходимая глубина рекурсии.

g.drawLine(a.x, a.y, b.x, b.y);

} else {

// находим длину отрезка (a; b).

double length = Math.pow(Math.pow(b.y - a.y, 2) + Math.pow(b.x - a.x, 2), 0.5);

// находим длину 1/3 отрезка (a; b)

double length1of3 = length / 3;

// находим т., делящую отрезок как 1:3.

Point a1 = new Point(a.x + (int) Math.round((length1of3 \* Math.cos(fi))),

a.y + (int) Math.round((length1of3 \* Math.sin(fi))));

// находим т., делящую отрезок как 2:3.

Point b1 = new Point(a1.x + (int) Math.round((length1of3 \* Math.cos(fi))),

a1.y + (int) Math.round((length1of3 \* Math.sin(fi))));

// находим т., которая будет вершиной равностороннего

// треугольника.

Point c = new Point(a1.x + (int) Math.round((length1of3 \* Math.cos(fi + Math.PI / 3))),

a1.y + (int) Math.round((length1of3 \* Math.sin(fi + Math.PI / 3))));

n--;

drawKochLine(g, a1, c, fi + Math.PI / 3, n);

drawKochLine(g, c, b1, fi - Math.PI / 3, n);

n--;

drawKochLine(g, a, a1, fi, n);

drawKochLine(g, b1, b, fi, n);

}

}

}