Ордена Трудового Красного Знамени

федеральное государственное бюджетное

образовательное учреждение высшего

образования «Московский технический

университет связи и информатики»

Кафедра информатики

Лабораторная работа № 4 «Fractal Explorer выбор и сохранение»

Выполнил:

Студент 2 курса

Факультета «ОТФ-2»

Группы БСТ-1602

Марченко Дмитрий

Вариант № 11

Москва, 2018

**Цель работы:** В этой задаче вы расширите генератор фракталов двумя новыми функциями. Во-первых, вы научитесь работать с множеством фракталов и сможете выбирать нужный вам фрактал с помощью выпадающего списка. Во - вторых, вы научитесь сохранять текущее изображения фрактала в файл.

**Задание:**

1. Создайте две новые имплементации генератора фракталов. Первый фрактал - tricorn, который должен быть в файле Tricorn.java. Как и раньше, вам следует создать подкласс FractalGenerator
2. Swing combo-boxes может управлять коллекцией объектов, но объекты должны предоставлять имплементацию to String().
3. Вам также следует добавить label в ваш пользовательский интерфейс перед выпадающим списком, что объяснит, для чего нужен выпадающий список. Вы можете это сделать, создав новый объект JPanel, добавив JLabel объект и JComboBOx
4. .И наконец, вам нужно добавить поддержку вашего выпадающего списка имплементации ActionListener. Вы можете проверить, является ли выпадающий список источником action-event и если да, вы можете извлечь выбранный объект из виджета и сделать его текущим генератором фракталов (вы можете использовать метод getSelectedItem())
5. добавить кнопку «Сохранить изображение» на ваш дисплей. Вы можете поставить обе кнопки «Сохранить» и «Сбросить» на вашу новую JPanel, а затем поставить эту панель в SOUTH часть фрейма, почти так же, как вы добавляли label и выпадающий список
6. Кнопка Сохранить также должна обрабатываться имплементацией ActionListener
7. нужно разрешить пользователю указывать, какой файл он хочет сохранить в изображение. Вы можете сделать это с классом javax.swing.JFileChooser, который упрощает задачу. Этот класс предоставляет метод showSaveDialog(), который дает всплывающее диалоговое окно «Сохранить файл», позволяя пользователю выбрать директорию для сохранения. Этот метод принимает графический компонент, который является родительским элементом выбора.
8. настроить выборщик файлов, чтобы он сохранял только png изображения, это пока единственный формат, с которым вы будете работать. вы можете это сделать с javax.swing.filechooser.FileNameExtensionFilter
9. Если пользователь успешно выбрал файл, следующим шагом является сохранение изображения фрактала на диск. Это может быть сложным, но Java уже включает в себя эту функцию. Класс javax.imageio.ImageIO позволяет реализовывать простые операции загрузки и хранения изображения. Вы можете использовать версию write(RenderedImage im, String formatName, File output. Формат будет png.
10. Конечно, вы заметите что метод write() может выдать исключение, поэтому вам нужно закрыть wrap запрос в блоке try/catch и разобраться с возможной ошибкой. Ваш блок catch должен проинформировать пользователя диалоговым окном. Опять же swing предоставляет класс javax.swing.JOptionPane чтобы упростить процесс создания информационных диалоговых окон или создания ввода да/нет.

**Анализ предметной области и инструментарий:** для разработки программы обработки информации, поступающей от пользователя в циклах и ее вывода взят пакетjava development kit-9.0.4 с классом Scanner для организации ввода и стандартный редактор тхт.

**Код программы:**

**Описание класса** **FractalExplorer:**

|  |
| --- |
|  |
|  |
|  | import java.awt.BorderLayout; |
|  | import javax.imageio.ImageIO; |
|  | import java.awt.\*; |
|  | import java.awt.event.\*; |
|  | import javax.swing.\*; |
|  | import java.awt.event.\*; |
|  | import java.io.\*; |
|  | import javax.swing.filechooser.FileNameExtensionFilter; |

|  |
| --- |
|  |
|  |
|  |
|  |
|  | public class FractalExplorer |
|  | { |
|  | public static void main(String[] args) |
|  | { |
|  | FractalExplorer MyExplorer = new FractalExplorer(500); //иниц. изобр в 500 пикс. |
|  | MyExplorer.createAndShowGUI(); //вызов метода интерфейса |
|  | MyExplorer.drawFractal(); //расчет фрактала |
|  | } |
|  | |  | | --- | |  | |  | private int MySize;  private int MyDispalySize; | |  | private JImageDisplay MyImage; | |  | private FractalGenerator MyGenerator; | |  | private FractalGenerator MyGeneratorT; | |  | private FractalGenerator MyGeneratorM; | |  | private FractalGenerator MyGeneratorB; | |  | private Rectangle2D.Double MyRange= new Rectangle2D.Double(0,0,0,0); | |  | private JComboBox Box =new JComboBox(); | |  | private Button B=new Button("Reset"); | |  | private Button S=new Button("Save"); | |
|  | FractalExplorer(int NewSize) |
|  | { //инициализация всех фракталов   |  |  | | --- | --- | |  | { | |  | MySize=NewSize; | |  | MyImage = new JImageDisplay(NewSize,NewSize); | |  | MyGenerator= new Tricorn(); | |  | MyGeneratorT= new Tricorn(); | |  | MyGeneratorM= new Mandelbrot(); | |  | MyGeneratorB= new BurningShip(); | |  | MyGenerator.getInitialRange(MyRange); | |  | } | |
|  |
|  |
|  |
|  |
|  | } |
|  | |  |  |  | | --- | --- | --- | |  |  |  | | //элементы упр. |  |  | public void createAndShowGUI()  { | |  |  |  | JFrame MyFrame =new JFrame(); | |  |  |  | MyFrame.setLayout(new BorderLayout()); | |  |  |  | JLabel MyLable=new JLabel("FRACTAL"); | |  |  |  | JPanel MyPanel=new JPanel(); | |  |  |  | JPanel MyPanel2=new JPanel(); | |  |  |  | MyPanel.add(MyLable); | |  |  |  | MyPanel.add(Box); | |  |  |  | MyFrame.add(MyPanel, BorderLayout.NORTH); | |  |  |  | Box.addItem(MyGeneratorT); | |  |  |  | Box.addItem(MyGeneratorM); | |  |  |  | Box.addItem(MyGeneratorB); | |  |  |  | MyPanel2.add(B); //кнопка сохр. | |  |  |  | MyPanel2.add(S); //кнопка ресет | |  |  |  | MyFrame.add(MyPanel2, BorderLayout.SOUTH); | |  |  |  | MyFrame.add(MyImage, BorderLayout.CENTER); | |  |  |  | MyFrame.setDefaultCloseOperation(MyFrame.EXIT\_ON\_CLOSE); | |  |  |  | MyFrame.pack(); | |  |  |  | MyFrame.setVisible(true); | |  |  |  | MyEventListener handler2=new MyEventListener(); | |  |  |  | MyFrame.setResizable(false); | |  |  |  | MouseMyAdapter handler=new MouseMyAdapter(); | |  |  |  | MyImage.addMouseListener(handler); | |  |  |  | B.addActionListener(handler2); | |  |  |  | S.addActionListener(handler2); | |  |  |  | Box.addActionListener(handler2); | |  |  |  | } | |
|  |  |
|  | public void drawFractal() //установка цвета каждого пикселя изображения |
|  | { |
|  | for(int x=0;x<MySize;x++) |
|  | { |
|  | for(int y=0;y<MySize;y++) |
|  | {  Поиск пискля для расчета при помощи мтолов в FractalGenerator |
|  | double X = FractalGenerator.getCoord(MyRange.x,MyRange.x+MyRange.width,MySize,x); |
|  | double Y = FractalGenerator.getCoord(MyRange.y,MyRange.y+MyRange.height,MySize,y); |
|  | int iteration = MyGenerator.numIterations(X,Y); |
|  | int ColoR=0; |
|  | if(iteration!=-1) //если пиксель не за пределами |
|  | { |
|  | float hue = 0.7f + (float)iteration/200f; //установка цвета согласно номеру последней итерации |
|  | ColoR = Color.HSBtoRGB(hue,1f,1f); |
|  | } |
|  | MyImage.drawPixel(x,y,ColoR); //покраска пикселя |
|  | } |
|  | } |
|  | MyImage.repaint(); //обновление изображения |
|  | } |
|  | public class MouseMyAdapter extends MouseAdapter //подкласс для получения //действий пользователя при помощи мыши   |  |  | | --- | --- | |  | { | |  | public void actionPerformed(ActionEvent e) | |  | { | |  | if(e.getSource().equals(B))//сравнение источника с //видом кнопки «ресет» | |  | { | |  | MyGenerator.getInitialRange(MyRange); | |  | } | |  | if(e.getSource().equals(S)) // “сохранение” | |  | { | |  | JFileChooser ChooseFile = new JFileChooser(); | |  | FileNameExtensionFilter FileFilter = new FileNameExtensionFilter("PNG Images","png"); //установка фильтра на //изображения типа png | |  | ChooseFile.setFileFilter(FileFilter); | |  | ChooseFile.setAcceptAllFileFilterUsed(false); | |  | if(ChooseFile.showSaveDialog(null)==JFileChooser.APPROVE\_OPTION) | |  | { | |  | try{ //сохранение файла в случае без ошибок | |  | File file = ChooseFile.getSelectedFile(); | |  | String PathM = file.getPath(); | |  | file=new File(PathM+ ".png"); | |  | ImageIO.write(MyImage.getImage(),"png",file); | |  | } | |  | catch(IOException err) | |  | { | |  | //вывод сообщения об ошибке JOptionPane.showMessageDialog(null,"error"); | |  | } | |  |  | |  | } | |  | } | |  | if(e.getSource().equals(Box))//источник список? | |  | { | |  | String H=Box.getSelectedItem().toString(); //выбранный //вариант и задание необходимого изображения | |  | if(H.equals("Mandelbrot")) | |  | { | |  | MyGenerator=MyGeneratorM; | |  | } | |  | if(H.equals("Tricorn")) | |  | { | |  | MyGenerator=MyGeneratorT; | |  | } | |  | if(H.equals("BurningShip")) | |  | { | |  | MyGenerator=MyGeneratorB; | |  | } | |  | MyGenerator.getInitialRange(MyRange); | |  | } | |  | drawFractal(); | |  | } | |  | } | |  | } | |

**Классы не подвергшиеся изменениям:**

FractalGenerator.java – обеспечивает обработку координат а также приближение.

Mandelbrot.java – обеспечивает расчет точек фрактала

Tricorn.java и BurningShip.java – производные от предыдущего имеют различные формулы расчета точек

**Выводы:** построены алгоритмы работы с объектами swing и работы со списками, сравнение элементов управления, обработки изображения, работы с интерфейсами и устройствами.