ФЕДЕРАЛЬНОЕ АГЕНСТВО СВЯЗИ

Ордена Трудового Красного Знамени

федеральное государственное бюджетное

образовательное учреждение высшего образования

«Московский Технический Университет Связи и Информатики» (МТУСИ)

Кафедра МКиИТ

Лабораторная работа №3 по дисциплине:

«Технологии программирования»

«Фрактал Эксплорер»

Выполнил:

Студент 2 курса

Группы БСТ-1602

Герасимов Алексей

Вариант №3

Москва, 2018

Оглавление:

[Цели и задачи: 3](#_Toc515869970)

[Анализ предметной области и выбор инструментария: 3](#_Toc515869971)

[Функции и их объяснение: 3](#_Toc515869973)

[Вывод: 8](#_Toc515869974)

Цели и задачи:

Соединить небольшое JAVA-приложение, которое может нарисовать некоторые фракталы.

Анализ предметной области и выбор инструментария:

В данной лабораторной работе я использовала бесплатный пакет Jdk, также стандартный редактор txt.

Функции и их объяснение:

1. *Создаём класс FractalGenerator.*

*getInitialRange(Rectangle2D.Double range)-метод позволяет фрактальному генератору определять какая часть комплексной плоскости является самой “интересной” для определённого фрактала.*

*numIterations-метод, реализующий итеративную функцию для фрактала.*

public class Mandelbrot extends FractalGenerator

{

public static int MAX\_ITERATIONS =2000;

public void getInitialRange(Rectangle2D.Double range)

{

range.x = -2;

range.y = -1.5;

range.width = 3;

range.height = 3;

}

public int numIterations(double x,double y)

{

int iter=0;

double re=0;

double im=0;

double moduleZ;

while(iter<MAX\_ITERATIONS)

{

double nextRe=re\*re-im\*im+x;

double nextIm=2\*re\*im+y;

moduleZ=nextRe\*nextRe+nextIm\*nextIm;

re=nextRe;

im=nextIm;

iter++;

if (moduleZ>4)

return iter;

}

return -1;

}

public String toString()

{

return "Mandelbrot";

}

}

1. *Создаём класс FractalExplorer.*

*createAndShowGUI()-метод инициализирующий GUI Swing- содержащий JImageDisplay, возращает и кнопку для сброса дисплея.*

*Включаем: rame.pack (); frame.setVisible(); frame.setResizable().*

*FractalGenerator.getCoord()-метод помощник, определяет координаты с плавающей точкой для определённого набора пиксельных координат.*

*Создаём внутренние классы: ActionListener(обработчик сбрасывает диапазон к начальному) и MouseAdapter.*

|  |
| --- |
| public class FractalExplorer{ |
|  |

|  |
| --- |
| private JImageDisplay display; |
|  |

|  |
| --- |
| private int dSize; |
|  |

|  |
| --- |
| private FractalGenerator fGenerator; |
|  |

|  |
| --- |
| private Rectangle2D.Double range = new Rectangle2D.Double(0,0,0,0); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public static void main(String[] args){ |
|  |

|  |
| --- |
| FractalExplorer expl = new FractalExplorer(800); |
|  |

|  |
| --- |
| expl.createAndShowGUI(); |
|  |

|  |
| --- |
| expl.drawFractal(); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* @param size - The height and width of the window |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| public FractalExplorer(int size){ |
|  |

|  |
| --- |
| dSize = size; |
|  |

|  |
| --- |
| display = new JImageDisplay(dSize, dSize); |
|  |

|  |
| --- |
| fGenerator = new Mandelbrot(); |
|  |

|  |
| --- |
| fGenerator.getInitialRange(range); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public void createAndShowGUI(){ |
|  |

|  |
| --- |
| // Create the Window |
|  |

|  |
| --- |
| JFrame frame = new JFrame("Fractal Explorer"); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| //Add fractal display |
|  |

|  |
| --- |
| display.addMouseListener(new mouse\_listener()); |
|  |

|  |
| --- |
| frame.add(display,BorderLayout.CENTER); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| //Add the button |
|  |

|  |
| --- |
| JButton button = new JButton("Reset"); |
|  |

|  |
| --- |
| button.addActionListener(new act\_listener()); |
|  |

|  |
| --- |
| frame.add(button, BorderLayout.SOUTH); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| // Set the default close action to exit |
|  |

|  |
| --- |
| frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| //The installation default settings |
|  |

|  |
| --- |
| frame.pack(); |
|  |

|  |
| --- |
| frame.setVisible(true); |
|  |

|  |
| --- |
| frame.setResizable(false); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public void drawFractal(){ |
|  |

|  |
| --- |
| for(int i=0;i<dSize;i++) |
|  |

|  |
| --- |
| for (int j=0;j<dSize;j++){ |
|  |

|  |
| --- |
| double xCoord = FractalGenerator.getCoord(range.x, range.x + range.width, dSize, i); |
|  |

|  |
| --- |
| double yCoord = FractalGenerator.getCoord(range.y, range.y + range.height, dSize, j); |
|  |

|  |
| --- |
| // Compute the number of iterations |
|  |

|  |
| --- |
| int iterations = fGenerator.numIterations(xCoord, yCoord); |
|  |

|  |
| --- |
| //If the color is not default |
|  |

|  |
| --- |
| if(iterations!=-1){ |
|  |

|  |
| --- |
| float hue = 0.7f + (float) iterations / 200f; |
|  |

|  |
| --- |
| int color = Color.HSBtoRGB(hue, 1f, 1f); |
|  |

|  |
| --- |
| display.drawPixel(i, j, color); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| // If color is default |
|  |

|  |
| --- |
| else display.drawPixel(i, j, 0); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| // Display the image on the screen |
|  |

|  |
| --- |
| display.repaint(); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| private class act\_listener implements ActionListener { |
|  |

|  |
| --- |
| public void actionPerformed(ActionEvent e) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| fGenerator.getInitialRange(range); |
|  |

|  |
| --- |
| display.clearImage(); |
|  |

|  |
| --- |
| drawFractal(); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| private class mouse\_listener extends MouseAdapter { |
|  |

|  |
| --- |
| public void mouseClicked(MouseEvent e) { |
|  |

|  |
| --- |
| // Get mouse's coordinates |
|  |

|  |
| --- |
| super.mouseClicked(e); |
|  |

|  |
| --- |
| int mouseX=e.getX(); |
|  |

|  |
| --- |
| int mouseY=e.getY(); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| double xCoord = FractalGenerator.getCoord(range.x, range.x + range.width, dSize, mouseX); |
|  |

|  |
| --- |
| double yCoord = FractalGenerator.getCoord(range.y, range.y + range.height, dSize, mouseY); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| fGenerator.recenterAndZoomRange(range, xCoord, yCoord, 0.5); |
|  |

|  |
| --- |
| display.clearImage(); |
|  |

|  |
| --- |
| drawFractal(); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

}

Вывод:

Мы увидели начальное представление фракталов, создали графический виджет.