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 №4 по дисциплине:

«Технологии программирования»

«Выбор и сохранение фракталов»

Выполнила:

Студентка 2 курса

Группы БСТ-1602

Фефелова Анастасия

Вариант №18

Москва, 2018

Оглавление

[Цели и задачи: 3](#_Toc515870352)

[Анализ предметной области и выбор инструментария: 3](#_Toc515870353)

[Функции и их объяснение: 3](#_Toc515870355)

[Выводы: 7](#_Toc515870357)

Цели и задачи:

Научиться работать с множеством фракталов и выбирать нужный с помощью выпадающего списка.

Анализ предметной области и выбор инструментария:

в данной лабораторной работе использованы бесплатный пакет Jdk.

Функции и их объяснение:

**Файл BurningShip.java** .

*Этот класс отвечает за расчёт фрактала BurningShip и есть только два отличия от класса Mandelbrot: математическая функция и возвращаемое значение toString().*

|  |
| --- |
| import java.awt.geom.Rectangle2D; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public class BurningShip extends FractalGenerator{ |
|  |

|  |
| --- |
| // Maximum iterations: |
|  |

|  |
| --- |
| public static final int MAX\_ITERATIONS = 2000; |
|  |

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

|  |
| --- |
| \* @param range - The range borders of the fractal |
|  |

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

|  |
| --- |
| public void getInitialRange (Rectangle2D.Double range){ |
|  |

|  |
| --- |
| range.x=-2; |
|  |

|  |
| --- |
| range.y=-2.5; |
|  |

|  |
| --- |
| range.width=range.height=4; |
|  |

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

|  |
| --- |
|  |
|  |

|  |
| --- |
| //Use formulas for fractal Mandelbrot |
|  |

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

|  |
| --- |
| \* @param x - The abscissa of the point |
|  |

|  |
| --- |
| \* @param y - The ordinate of the point |
|  |

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

|  |
| --- |
| public int numIterations(double x, double y){ |
|  |

|  |
| --- |
| double re=0; |
|  |

|  |
| --- |
| double im=0; |
|  |

|  |
| --- |
| for (int i=0; i<MAX\_ITERATIONS; i++){ |
|  |

|  |
| --- |
| double re1=re\*re-im\*im+x; |
|  |

|  |
| --- |
| double im1=Math.abs(2\*re\*im)+y; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| //If we have reached the Mandelbrot condition, we will return the number of iterations |
|  |

|  |
| --- |
| if ((im\*im+re\*re) > 4) |
|  |

|  |
| --- |
| return i; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| //The change the real and imaginary parts for the next iteration |
|  |

|  |
| --- |
| re=re1; |
|  |

|  |
| --- |
| im=im1; |
|  |

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

|  |
| --- |
| return -1; |
|  |

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

|  |
| --- |
| public String toString (){ |
|  |

|  |
| --- |
| return "BurningShip"; |
|  |

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

|  |
| --- |
|  |
|  |

}

**Класс act\_listener.**

*Класс act\_listener способствует открытию нового фрактала, а также сохранению текущего изображения.*

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

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

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

|  |
| --- |
| String command = e.getActionCommand(); |
|  |

|  |
| --- |
| if (command.equals("Reset")){ |
|  |

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

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

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

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

|  |
| --- |
|  |
|  |

|  |
| --- |
| if (command.equals("Save")){ |
|  |

|  |
| --- |
| JFileChooser chooser = new JFileChooser(); |
|  |

|  |
| --- |
| FileFilter filter = new FileNameExtensionFilter("PNG Images", "png"); |
|  |

|  |
| --- |
| chooser.setFileFilter(filter); |
|  |

|  |
| --- |
| chooser.setAcceptAllFileFilterUsed(false); |
|  |

|  |
| --- |
| if(chooser.showSaveDialog(frame) == JFileChooser.APPROVE\_OPTION){ |
|  |

|  |
| --- |
| try{ |
|  |

|  |
| --- |
| ImageIO.write(display.getImage(), "png", chooser.getSelectedFile()); |
|  |

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

|  |
| --- |
| catch(IOException ex){ |
|  |

|  |
| --- |
| JOptionPane.showMessageDialog (frame, ex.getMessage(), "Unable to save image", JOptionPane.ERROR\_MESSAGE); |
|  |

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

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

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| return; |
|  |

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

|  |
| --- |
|  |
|  |

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

|  |
| --- |
|  |
|  |

|  |
| --- |
| if ((JComboBox)e.getSource()==box){ |
|  |

|  |
| --- |
| fGenerator = (FractalGenerator)box.getSelectedItem(); |
|  |

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

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

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

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

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

}

**Файл Tricorn.java.**

*Класс фрактала, как и Mandelbrot или BurningShip.*

|  |
| --- |
| import java.awt.geom.Rectangle2D; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public class Tricorn extends FractalGenerator{ |
|  |

|  |
| --- |
| // Maximum iterations: |
|  |

|  |
| --- |
| public static final int MAX\_ITERATIONS = 2000; |
|  |

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

|  |
| --- |
| \* @param range - The range borders of the fractal |
|  |

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

|  |
| --- |
| public void getInitialRange (Rectangle2D.Double range){ |
|  |

|  |
| --- |
| range.x=-2; |
|  |

|  |
| --- |
| range.y=-2; |
|  |

|  |
| --- |
| range.width=range.height=4; |
|  |

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

|  |
| --- |
|  |
|  |

|  |
| --- |
| //Use formulas for fractal Mandelbrot |
|  |

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

|  |
| --- |
| \* @param x - The abscissa of the point |
|  |

|  |
| --- |
| \* @param y - The ordinate of the point |
|  |

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

|  |
| --- |
| public int numIterations(double x, double y){ |
|  |

|  |
| --- |
| double re=0; |
|  |

|  |
| --- |
| double im=0; |
|  |

|  |
| --- |
| for (int i=0; i<MAX\_ITERATIONS; i++){ |
|  |

|  |
| --- |
| double re1=re\*re-im\*im+x; |
|  |

|  |
| --- |
| double im1=-2\*re\*im+y; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| //If we have reached the Mandelbrot condition, we will return the number of iterations |
|  |

|  |
| --- |
| if ((im\*im+re\*re) > 4) |
|  |

|  |
| --- |
| return i; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| //The change the real and imaginary parts for the next iteration |
|  |

|  |
| --- |
| re=re1; |
|  |

|  |
| --- |
| im=im1; |
|  |

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

|  |
| --- |
| return -1; |
|  |

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

|  |
| --- |
| public String toString (){ |
|  |

|  |
| --- |
| return "Tricorn"; |
|  |

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

}

Выводы:

Получен навык исследования различных фракталов и сохрания их на диск.