

Московский Государственный Технический Университет
имени Н.Э. Баумана

Факультет ИУ «Информатика и системы управления»

Кафедра ИУ-3 «Информационные системы и телекоммуникации»

Отчет к лабораторной работе №5
по курсу «Технологии программирования»
“Углубленное ознакомление работы классов в языке программирования Java”

Продолжительность работы: 4 ак. часа

Сдал

Ситниченко С. А. ИУ3-42Б

Принял

Руденкова Ю.С.

Москва, 2022

Лабораторная №5

Тема: Программа для изучения фракталов

Цель работы: Добавить две новые функции к программе Fractal Explorer. Во первых, возможность отображения нескольких фракталов и выбор фрактала с помощью выпадающего списка. Во вторых, возможность сохранить текущее изображение фрактала в файл. В верхней части окна программы Fractal Explorer теперь будет два виджета для выбора фрактала, а внизу добавится кнопка "Save" для сохранения текущего изображения фрактала. Так как теперь имеется несколько источников событий, вы сможете практически изучить вопрос обработки событий от всех этих источников в одном ActionListener в вашем классе.

Фрактал вида Tricorn

```
public class Tricorn extends FractalGenerator{
    1 usage
    public static final int MAX_ITERATIONS = 2000;
    3 usages
    public void getInitialRange(Rectangle2D.Double range) {
        range.x = -2;
        range.y = -2;
        range.width = 4;
        range.height = 4;
    }
    1 usage
    public int numIterations(double x, double y){
        double z = 0;
        double zi = 0;
        for(int i = 0; i < MAX_ITERATIONS; i++){
            double zrealup = z * z - zi * zi + x;
            double zimup = -1 * (2 * z * zi + y);
            z=zrealup;
            zi=zimup;
            if(z * z + zi * zi > 4){
                return i;
            }
        }
        return -1;
    }
    @Override
    public String toString() { return "Tricorn"; }
}
```

Рис. 1 Tricorn.java

Фрактал вида Burning ship

```
public class BurningShip extends FractalGenerator{
    1 usage
    public static final int MAX_ITERATIONS = 2000;
    3 usages
    public void getInitialRange(Rectangle2D.Double range){
        range.x = -2;
        range.y = -2.5;
        range.width = 4;
        range.height = 4;
    }
    1 usage
    public int numIterations(double x, double y){
        double z = 0;
        double zi = 0;
        for(int i = 0; i < MAX_ITERATIONS; i++){
            double zrealup = z * z - zi * zi + x;
            double zimup = 2 * Math.abs(z * zi) + y;
            z = zrealup;
            zi = zimup;
            if(z * z + zi * zi > 4){
                return i;
            }
        }
        return -1;
    }
    @Override
    public String toString() { return "BurningShip"; }
}
```

Рис. 2 BurningShip.java

Сохранение

```
private class ButtonHand implements java.awt.event.ActionListener{
    public void actionPerformed(ActionEvent e){
        String command = e.getActionCommand();
        if(command.equals("Reset")){
            freak.getInitialRange(area);
            drawFractal();
        }
        if(command.equals("Save")){
            JFileChooser chooser = new JFileChooser();
            FileFilter filter = new FileNameExtensionFilter( "description: \"PNG Images\", ...extensions: \"png\"");
            FileFilter filter1 = new FileNameExtensionFilter( "description: \"jpg\", ...extensions: \"jpg\"");
            FileFilter filter2 = new FileNameExtensionFilter( "description: \"gif\", ...extensions: \"gif\"");
            chooser.addChoosableFileFilter(filter);
            chooser.addChoosableFileFilter(filter1);
            chooser.addChoosableFileFilter(filter2);
            chooser.setAcceptAllFileFilterUsed(false);
            chooser.showSaveDialog(frame);
            System.out.println(chooser.getFileFilter().toString());
            try{
                ImageIO.write(myScreen.image, formatName: "png", chooser.getSelectedFile());
            }
            catch(Exception u){
                JOptionPane.showMessageDialog(frame, u.getMessage(), title: "ОшибкаСохраненияФайла",JOptionPane.ERROR_MESSAGE);
            }
        }
        if(e.getSource()==boxList){
            freak = (FractalGenerator)boxList.getSelectedItem();
            freak.getInitialRange(area);
            drawFractal();
        }
    }
}
```

Рис. 3 Обработка сохранения в обработчике кнопки

Результат

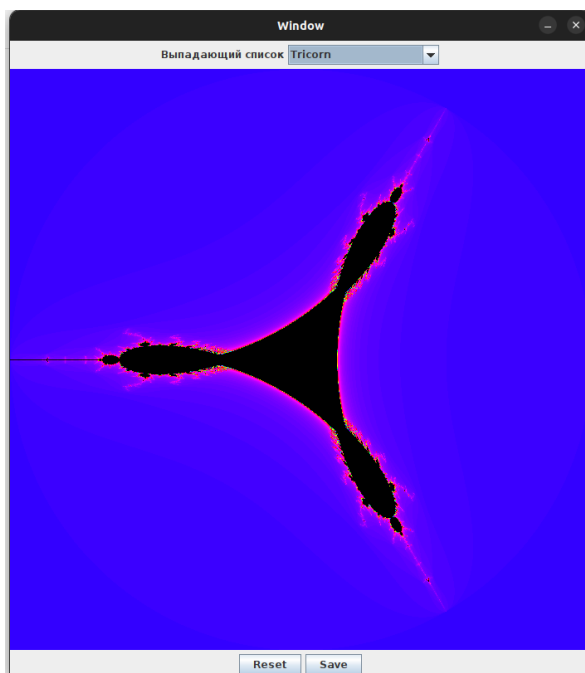


Рис. 4 результат Tricorn

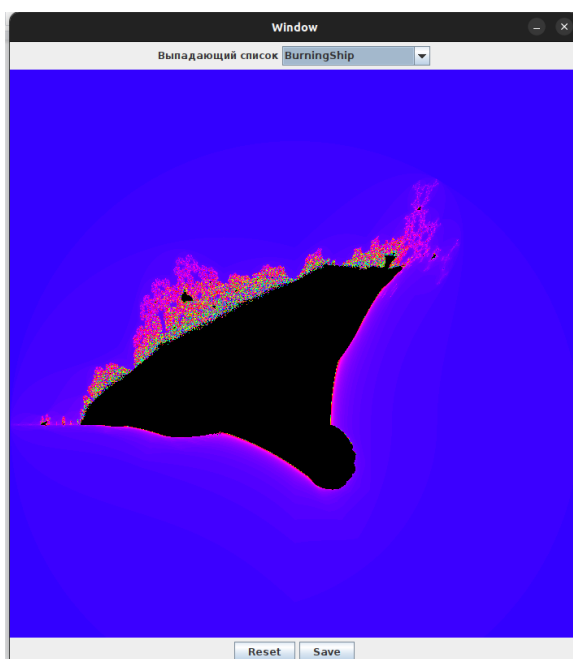


Рис. 5 результат BurningShip

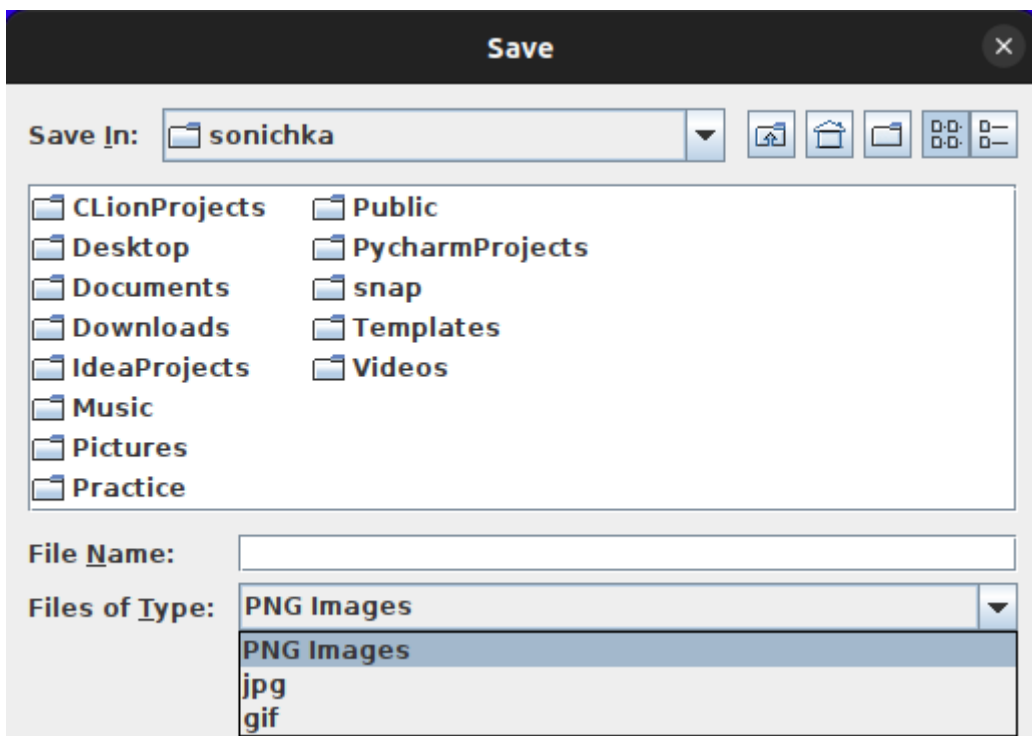


Рис. 6 доп. задание

Вывод:

В ходе лабораторной работы мы реализовали изображение нескольких фракталов и сохранения фрактала в файл. Также мы добавили обработку событий выпадающего списка и сохранение изображения на диск.