**КУРСОВАЯ РАБОТА**

**Реализация построения множества Мандельброта и представление его в виде картинки**

по дисциплине «Языки и средства функционального программирования»

Выполнил

студент гр. 3530904/80002 Протасов М. А.

Руководитель Лукашин А. А.

"17" декабря 2019 г.

Оглавление

[Описание задачи 3](#_Toc27926067)

[Описание решения 3](#_Toc27926068)

[Скриншот работы программы 3](#_Toc27926069)

[Вывод 3](#_Toc27926070)

[Приложение 4](#_Toc27926071)

[Main 4](#_Toc27926072)

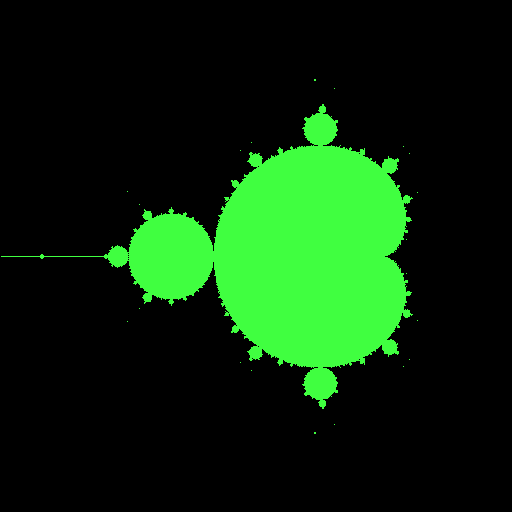
# Описание задачи

Реализовать построение множества Мандельброта и представления его в виде картинки.

# Описание решения

Используем стандартный алгоритм построения множества Мандельброта. Если в какой-то момент вычислений точка отойдет от начала координат на расстояние 2, то мы можем принять, что она уйдет в бесконечность.

# Скриншот работы программы



# Вывод

В ходе выполнения данной курсовой работы были получены навыки работы с языком функционального программирования Scala и навыки работы с библиотеками для вывода изображения из языка Java

# Приложение

## **Main**

import java.awt.Color  
import java.awt.image.BufferedImage  
import java.io.File  
import javax.imageio.ImageIO  
  
object Main {  
  
 class Complex (val real : Double, val imaginary : Double) {  
 def +(c : Complex) = new Complex(real + c.real, imaginary + c.imaginary)  
 def \*(c : Complex) = new Complex(real \* c.real - imaginary \* c.imaginary, real \* c.imaginary + imaginary \* c.real)  
 def abs() = real\*real + imaginary\*imaginary  
 }  
  
 def main(args: Array[String]) {  
 val size = 512;  
 val img = new BufferedImage(size, size, BufferedImage.*TYPE\_INT\_RGB*)  
 val graphics = img.getGraphics()  
  
 for (y0 <- 0 to size) {  
 for (x0 <- 0 to size) {  
 val a = x0 \* 3.0 / size - 2.0  
 val b = y0 \* 3.0 / size - 1.5  
 var z = new Complex(0,0)  
 val c = new Complex(a, b)  
 var count = 0  
 while (z.abs() < 4 && count < size) {  
 z = z \* z + c  
 count += 1  
 }  
 if (z.abs() < 4){  
 graphics.setColor(new Color(64, 255, 64))  
 graphics.drawLine(x0, y0, x0, y0)  
 }  
 }  
 }  
 ImageIO.*write*(img, "png", new File("Mandel.png"))  
 }  
}