КУРСОВая Работа

Создание приложения на языке Scala на тему

«Построение множества Мандельборта»

по дисциплине «Функциональное программирование»

Выполнил

студент гр.3530904/80004 < > Краснопёров Н.М.

Руководитель <> Лукашин А.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2019 г.

Санкт-Петербург

2019

**Оглавление**

[Оглавление 2](#_Toc27510388)

[Задание 2](#_Toc27510389)

[Результат работы программы 2](#_Toc27510390)

[Программа 3](#_Toc27510391)

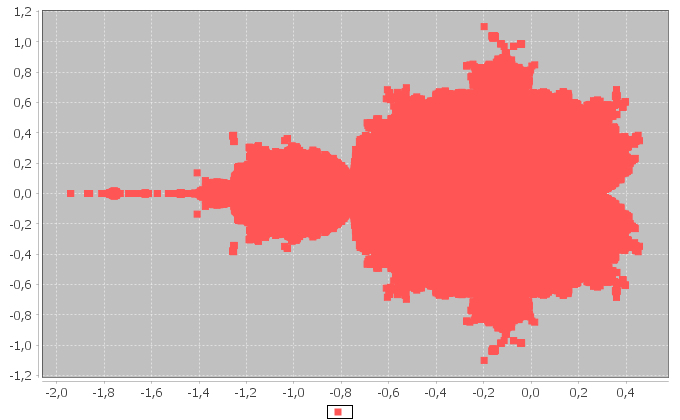
[Описание алгоритма решения 4](#_Toc27510392)

[Заключение 4](#_Toc27510393)

# **Задание**

Построить множество Мандельборта и представить его в виде картинки (без использования actors (akka))

# **Результат работы** **программы**



# 

# **Программа**

**Main.scala**

package main

import scalax.chart.api.\_

import scala.collection.mutable.ListBuffer

import complex.\_

object Main {

def main(args: Array[String]) {

val z = new Complex(0, 0)

val data = mandelbort(ListBuffer())

val title: String = "Mandelbort"

val chart = XYLineChart(data)

chart.plot.setRenderer(new org.jfree.chart.renderer.xy.XYLineAndShapeRenderer(false, true))

chart.saveAsPNG("mandelbort.png")

}

def mandelbort(container: ListBuffer[(Double, Double)]): List[(Double, Double)] = {

for (re <- -2.0 until 2.0 by 0.001)

for (im <- 2.0 until -2.0 by -0.001)

if (haslim(new Complex(0.0, 0.0), new Complex(re, im), 0))

container += ((re, im))

container.toList

}

def haslim(z: Complex, c: Complex, count: Int): Boolean =

{

if (z.module < 4 && count < 100)

haslim(z \* z + c, c, count + 1)

else if (count == 100)

true

else

false

}

}

**Complex.scala**

package complex

class Complex(val real: Double, val image: Double) {

override def toString = real + (if (image < 0) "-" + -image else "+" + image) + "\*i"

def +(rhs: Complex) = new Complex(rhs.real + real, rhs.image + image)

def \*(rhs: Complex) = new Complex(real \* rhs.real - image \* rhs.image, image \* rhs.real + real \* rhs.image)

def module: Double = real \* real + image \* image

}

# **Описание алгоритма решения**

1. Был создан класс Complex для упрощения совершения операций над комплексными числами.
2. Была подключена библиотека chart для графического изображения нашего множества
3. Создана функция mandelbort для перебора всех точек в диапазоне (-2;2) по оси ox и (2;-2) по оси oy(в этом диапазоне лежат все необходимые нам точки).
4. Каждая точка проверяется функцией haslim нам наличие предела (все элементы данного рекуррентного соотношения не должны превышать по модулю 4), функция является рекурсивной, для простоты вычислений выбраны первые 100 элементов, в идеале все элементы, имеющие натуральные индексы, этого соотношения должны не превышать по модулю 4.
5. Полученные точки записываются в список, затем мы закрашиваем полученные точки на графике и сохраняем его в файл mandelbort.png.

# **Заключение**

В данной курсовой работе был использован язык функционального программирования Scala для графического представления множества Мандельборта. Также были использованы сторонние библиотеки для визуализации множества. Использование данного языка позволяет нам выполнить большое количество вычислений за относительно короткий промежуток времени.