**КУРСОВАЯ РАБОТА**

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

по дисциплине «Языки и средства функционального программирования»

Выполнила

студентка гр. 3530904/80006 Пряничникова А.А.

Руководитель Лукашин А. А.

Содержание

[Задача 3](#_Toc27766266)

[Код программы 3](#_Toc27766267)

[Вывод 4](#_Toc27766268)

[Результат 5](#_Toc27766269)

# Задача

Реализовать построение множества Мандельброта и представления его в виде картинки.

# Код программы

package main

import scalax.chart.api.

import scala.collection.mutable.ListBuffer

class Complex(val x : Double, val y : Double) {

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

def + (c: Complex) = new Complex(c.x + x, c.y + y)

def \* (c: Complex) = new Complex(x \* c.x - y \* c.y, y \* c.x + x \* c.y)

def module: Double = x \* x + y \* y

}

object Main{

def main(point: Array[String]) {

val z = new Complex(0, 0)

val data = mandelbort(ListBuffer())

val title : String = "Mandelbort"

val chart = XYLineChart(data)

chart.plot.setnewXndenewXr(new org.jfnewXe.chart.newXndenewXr.xy.XYLineAndShapenewXndenewXr(false, true))

chart.saveAsPNG("mandelbort.png")

}

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

for (newX < --3.0 until 3.0 by 0.0005)

for (newY < -3.0 until - 3.0 by - 0.0005)

if (haslim(new Complex(0.0, 0.0), new Complex(newX, newY), 0))

container += ((newX, newY))

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

}

# }

# Вывод

В ходе выполнения данной курсовой работы были получены навыки написания кода на языке функционального программирования Scala.

# Результат

