КУРСОВая Работа

Создание приложения на языке Scala на тему

«Построение множества Мандельборта»

по дисциплине «Функциональное программирование»

Выполнил студент гр.3530904/80001 Неупокоев Д.А.

Руководитель  Лукашин А.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2019 г.

Санкт-Петербург

2019

**Оглавление**

[Введение 2](#__RefHeading___Toc2352_4129578054)

[Задание 2](#__RefHeading___Toc2292_4129578054)

[Ход работы 2](#__RefHeading___Toc2330_4129578054)

[Результат работы программы 3](#__RefHeading___Toc2339_4129578054)

[Приложения 4](#__RefHeading___Toc2296_4129578054)

# **Введение**

Функциона́льное программи́рование — раздел дискретной математики и парадигма программирования, в которой процесс вычисления трактуется как вычисление значений функций в математическом понимании последних (в отличие от функций как подпрограмм в процедурном программировании).

# **Задание**

Построить множество Мандельборта и представить его в виде картинки

# **Ход работы**

1. Установил Scala SBT
2. Установил библиотеку Chart
3. Написал файл для сборки build.sbt
4. Написал класс комплексных чисел в Complex.scala
5. Написал основную часть программы в Main.scala
6. Собрал программу командой «sbt run» и получил результат в файле Results.png

# **Результат работы** **программы**

# **Вывод**

В результате работы была реализованна программа для представления множества Мандельборта в виде картинки.

# **Приложения**

**Main.scala**

package main

import scalax.chart.api.\_

import scala.collection.mutable.ListBuffer

import complex.\_

object Main {

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 < 300)

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

else if (count == 300)

true

else

false

}

def main(args: Array[String]) {

val z = new Complex(0, 0)

val data = mandelbort(ListBuffer())

val title: String = "Results"

val chart = XYLineChart(data)

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

chart.saveAsPNG("Results.png")

}

}

**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

}

**build.sbt**

name := "main"

version := "0.1"

scalaVersion := "2.11.8"

libraryDependencies += "com.github.wookietreiber" %% "scala-chart" % "latest.integration"