**Индивидуальное домашнее задание № 1**

1. **Определение погрешностей**

Найти абсолютную и относительную погрешности выражения , где у заданных чисел все цифры верные. Полученный результат представить с максимальным количеством верных цифр.

**Задание**

**Решение**

Программа выполнена на языке программирования Scala.

import java.lang

import scala.math.\_;

/\*\*

\* Created with IntelliJ IDEA.

\* User: SabLeOn

\*/

class Number( val value:Double,

val a\_error:Double,

val r\_error:Double) {

def this(const:Int) = this(const, 0,0)

def this(v:Double, a:Double) = this(v, a, a / v)

def +(that:Number) = new Number(value + that.value,

a\_error + that.a\_error,

(a\_error + that.a\_error) / (value + that.value)

)

def -(that:Number) = new Number(value - that.value,

a\_error + that.a\_error,

(a\_error + r\_error) / (value - that.value)

)

def \*(that:Number) = new Number(value \* that.value,

abs(value) \* that.a\_error + abs(that.value) \* a\_error + a\_error \* that.a\_error,

r\_error + that.r\_error + r\_error \* that.r\_error

)

def \*(c:Int, x:Number) = new Number(c \* value, a\_error, a\_error / c \* value)

def /(that:Number) = new Number(value / that.value,

(abs(value) \* that.r\_error + abs(that.value) \* r\_error) /

((that.value - that.a\_error) \* abs(that.value)),

(r\_error + that.r\_error) / (1 - that.r\_error)

)

def isNaN:Boolean = value.isNaN

override def toString = "%.3f(\u00b1%.4f)(1\u00b1%.4f)".format(

new lang.Double(value), new lang.Double(abs(a\_error)), new lang.Double(abs(r\_error)))

}

object Number {

implicit def int2number(i: Int) = new Number(i)

}

object ErrorEstimation {

val a:Number = new Number(0.035, 0.001)

val b:Number = new Number(7.05, 0.01)

val c:Number = new Number(-2.76, 0.01)

def sqrt(x: Number): Number = new Number(math.sqrt(x.value), x.a\_error, x.a\_error / math.sqrt(x.value))

def f(a:Number, b:Number, c:Number):Number = {

val num:Number = a \* b - (a \* sqrt((3 \* a - c) / (2 \* b)))

val den:Number = a \* a - b \* b

val result:Number = num / den

if (!result.isNaN) result

else throw new IllegalArgumentException("numerator = %s\ndenominator = %s".format(num, den))

}

def main(args: Array[String]):Unit = try

println("f(a,b,c) = %s".format(f(a, b, c)))

catch {

case e: IllegalArgumentException => println(e.getMessage)

}

}

**Результат работы программы**

f(a,b,c) = -0,005(±0,0033)(1±0,1607)

Значение функции: -0.005, абсолютная погрешность: ±0,0033, относительная погрешность: 16,07%