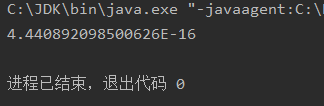
Scala编程语言 作业及练习

1. 在Scala REPL中，计算3的平方根然后再对该值求平方。现在，这个结果与3相差多少？

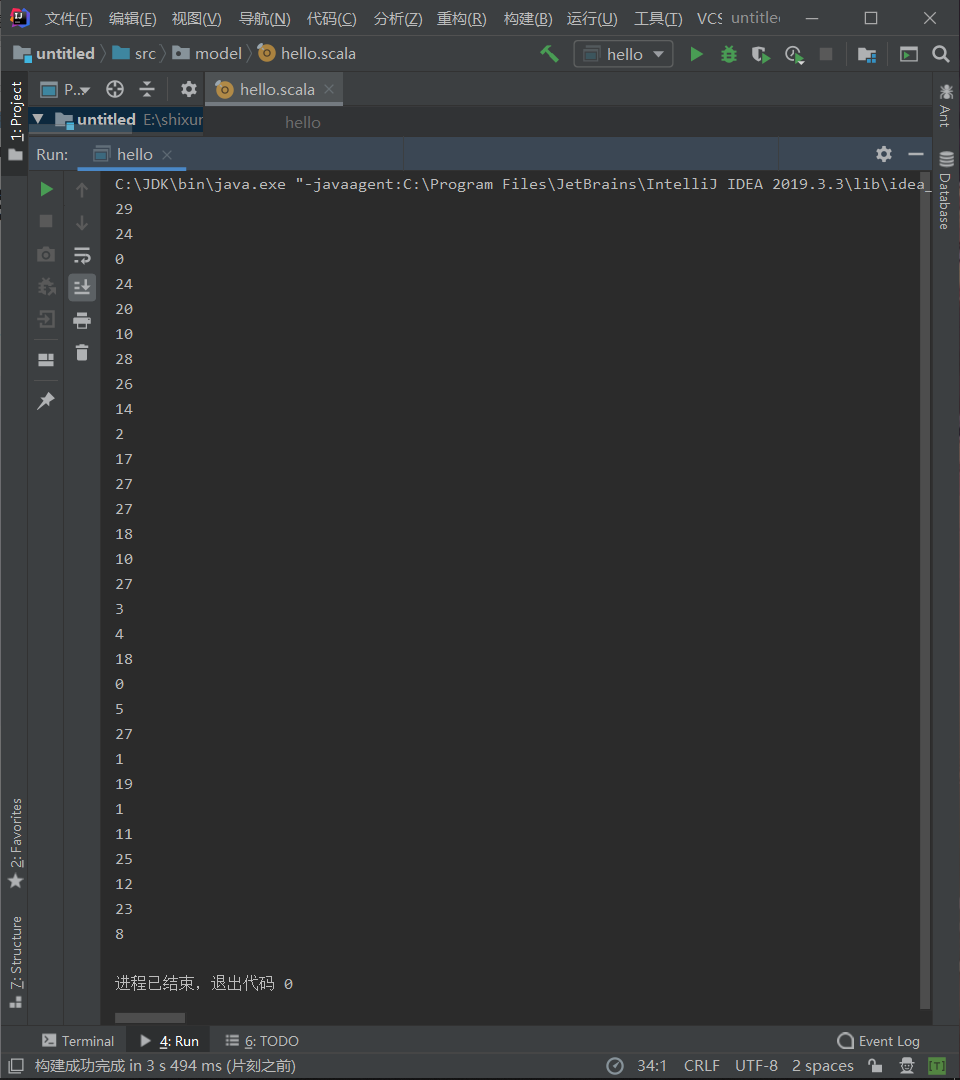


1. 变量声明时val和var有什么区别？

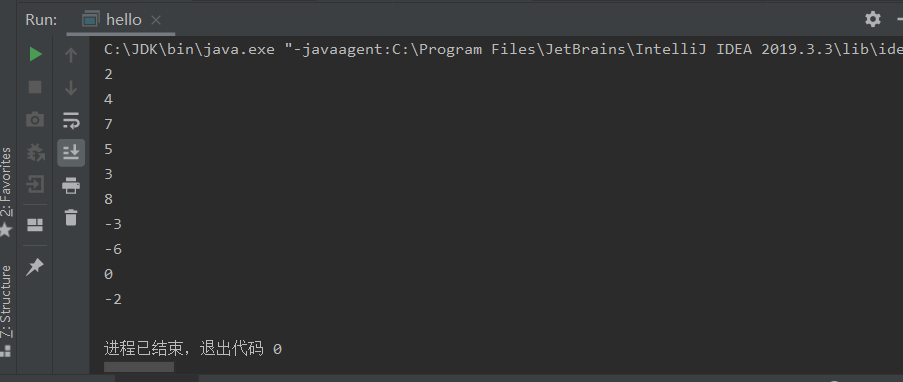
val 表示该变量是一个不可变量，相当于 java 中加了 const 限定词的作用。

var 表示该变量是一个可变变量，可以对变量的值进行修改

1. 编写一段代码，将sum设置一个30个随机整数的数组，要求随机数介于10和n之间



1. 给定一个整数数组，产出一个新的数组，包含元数组中的所有正值，以原有顺序排列，之后的元素是所有零或负值，以原有顺利排列。数组中的值为：-3,2,4,-6,7,5,0,3,8,-2



1. 一个数字如果为正数，则它的signum为1;如果是负数,则signum为-1;如果为0,则signum为0.编写一个函数来计算这个值

def signum (x: Int): Int = // signum: signum[](val x: Int) => Int

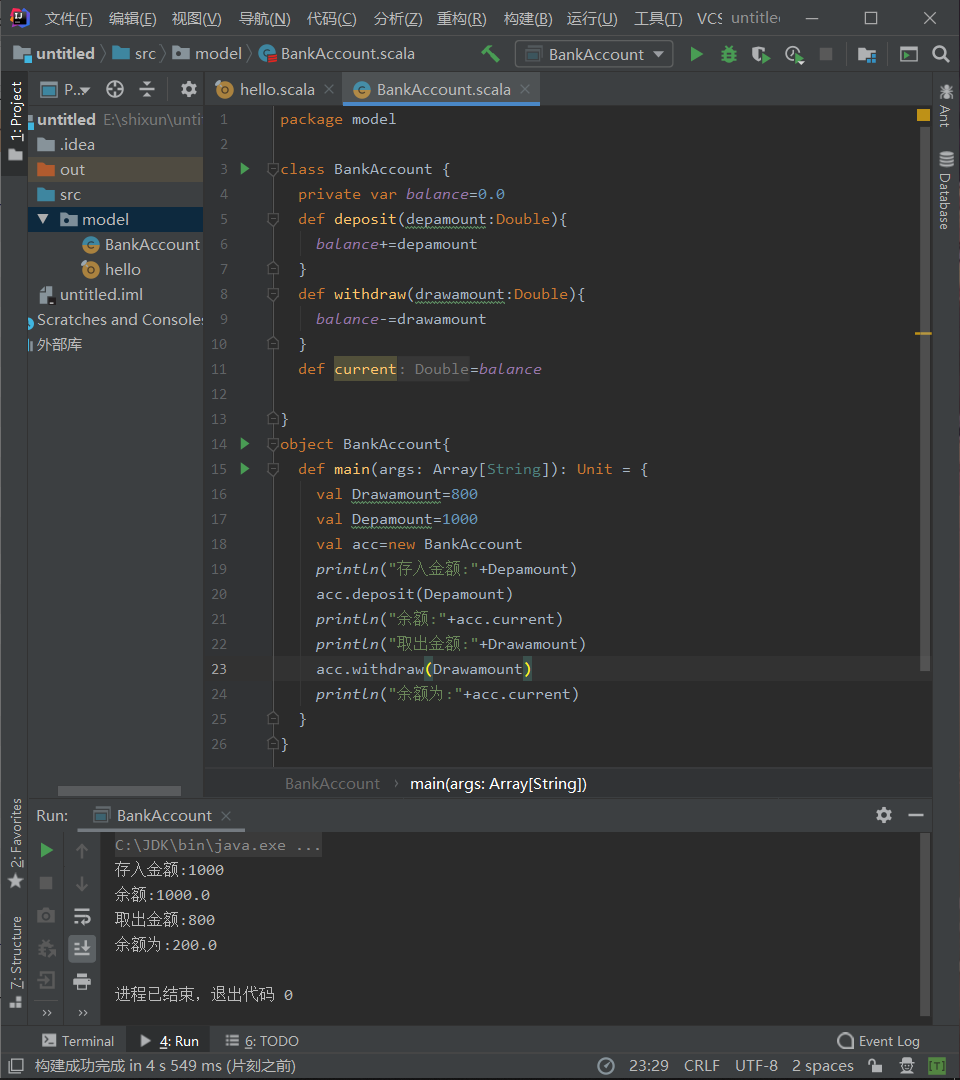
if (x > 0) 1

else if (x < 0) -1

else 0

signum(1) //res0: Int = 1

1. 编写一个BankAccount类，加入deposit和withdraw方法，和一个只读的balance属性



1. 定义一个Point类和一个半生对象，使得我们可以不用new而直接用Point（3,4）实例apply方法的使用

class Point(x:Int,y:Int){

override def toString:String = " x="+x+" y="+y

}

object Point extends App{

def apply(x:Int,y:Int) = {

new Point(x,y)

}

val p = Point(3,4)

println(p)

}

1. 编写一个扑克牌4种花色的枚举，让其toString方法分别返回不同图标.(图标可以用数字或字母来代替)

object Card extends Enumeration with App{

val M = Value("♣")

val T = Value("♠")

val H = Value("♥")

val F = Value("♦")

println(Card.M)

println(Card.T)

println(Card.H)

println(Card.F)

}

1. 设计一个point类，其x和y坐标可以通过构造器提供。提供一个子类LabeledPoint,其构造器接受一个标签值和x,y坐标，比如：new Labeledpoint（"Black Thursday”,1929,230,07）

class Point(x:Int,y:Int){  
}  
  
class LabeledPoint(label:String,x:Int,y:Int) extends Point(x,y){  
}

1. 编写一个过程countdown(n:Int)，打印从n到0的数字

