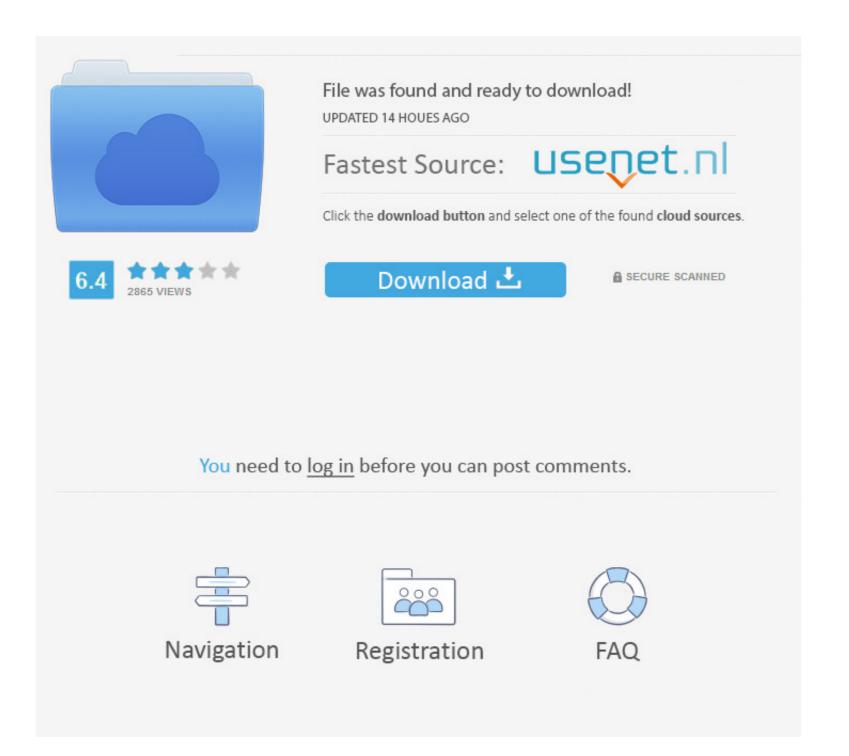


Introduction To The Art Of Programming Using Scala Pdf

1/4

```
scala> 1+2
rt java.awt.{Color,Font,Graphics2D}
rt java.awt.image.BufferedImage
 : swing.
: javax.īmageio.ImageIO
: java.io.File
T RayTrace {
    main(args : Array[String]) : Unit = {
    val ray=new Ray(new Point(0,-2,0), new Point(0,-1,0))
    val imgenew BufferedImage: 600,800,8ufferedImage.TYPE_INT_ARGB)
    val lights=List(new AmbientLight(new FColor(.1f..1f..1f,.1f)),
        new DirectionLight(new FColor(0.75f,0.75f,0.75f,1), new Vect(1,1,-1)))
    val scene=new OttreeScene(new Point(0,0,0),20)
    val Scene=new OttreeScene(new Point(0,0,0),20)
    val dTexthopkCover(scene)
                                                                                                                                              res0: Int = 3
             over(scene)
st(0,-4,0),new Point(-0.5,-3,-0.5),new Vect(1,0,0),
    scala> "Welcome to Scala"
                                                                                                                                              res1: java.lang.String = Welcome to Scala
    ildTextbookCover(scene:Scene) {
:lear = new Color(0,0,0,0)
:tairs = new GeomCylinder(Point(0,0,-0.98),Point(0,0,1.02),0.5,
    => {
val ang = math.Pi-math.atan2(p.x,p.y)
if((-4 to 4 by 2).exists(i => ((p.z*2.9*i)*math.Pi-ang).abs<2)) Color.red else clear
    val ang = math.Pi-math.atan2(p.x,p.y)
if((-4 to 4 by 2).exists(i => ((p.z*2.9+i)*math.Pi-ang).abs<2)) 0.4 else 0.6</pre>
     addGeom stairs
addGeom new GeomPolyFunc{Array(Point(-300,300,-1),Point(-300,-3,-1),Point(300,-3,-1),Point(300,300,-1)},
>> Vect(6,0,1),
>> Color.black,
      > U.a)
ddGeom fileTextPoly(
ome/mlevis/workspace/ScalaBock/src/scalabook/adt/TreeMap.scala",
nt[1.3.2.2.].2.clea
                                                                                                                                                                                                                                                                                                                        package scalabook.adt
         /mlewis/workspace/ScalaTest2_8/RayTrace.scala",
/2.5,1,3),2,clear!
                                                                                                                                                                                                                                                                                                                        import collection.mutable
         Vmlewis/Documents/Teaching/ScalaBook/LatexVersion/Chapters/Functions/Code/Factorial.scala",
-0.7,-2,-0.75),2,clear)
                                                                                                                                                                                                                                                                                                                        import annotation.__
   le addGeom fileTextPoly(
"/home/mlewis/workspace/ScalaTest2_8/repl.txt",
Point(-0.6,-0.5,1.75),2,clear)
                                                                                                                                                                                                                                                                                                                        class TreeMap[K,V](comp:(K,K)=>Int) extends mutable.Map[K,V] {
  ene addGeom fileTextPoly(
"/home/mlewis/workspace/ScalaTest2_8/dot.scala",
Point(-0.2,-1.2,0),2,clear)
                                                                                                                                                                                                                                                                                                                           private class Node(var key:K, var data:V) {
                                                                                                                                                                                                                                                                                                                              var left:Node = null
                                                                                                                                                                                                                                                                                                                               var right:Node = null
 texturePanel(p:Point,r:Vect,d:Vect,n:Vect,ref:Double,fname:String):Geometry = {
    l pnts=Array(p·r*6.5,p+d-r*6.5,p+d+r*6.5,p+r*6.5)
    l texture=new TextureColorfunc(ImageIO.read(new File(fname)),p-r*6.5,r,d)
    w GeomPolygon(pnts,Array(n,n,n,n),(x)=>texture(x),Array(ref,ref,ref,ref))
f render(eye:Point,topLeft:Point,right:Vect,down:Vect,img:BufferedImage,geom:Geometry,lights:List[Light],numRays:Int) {
for(i <- 0 until img.getWidth par; j <- 0 until img.getHeight) {
   img.setRGB(i,img.getHeight-j-l,{((l to numRays).map(index=>{
      val ray=new Ray(eye,topLeft+right*(i+(if(index>0) math.random else 0))/img.getWidth+down*{j+(if(index>0) math.random else 0})/img.getHeight)
   castRay(ray,geom,lights,0)
  }).reduceLeft(_+_})/numRays).toColor.getRGB)
                                                                                                                                                                                                                                                                                                                           private var root:Node = null
                                                                                                                                                                                                                                                                                                                            def +=(kv:(K,V)) = \{
                                                                                                                                                                                                                                                                                                                               if(root==null) {
                                                                                                                                                                                                                                                                                                                                  root = new Node(kv. 1,kv. 2)
                                                                                                                                                                                                                                                                                                                               } else {
vate def castRay(ray:Ray,geom:Geometry,lights:List[Light],cnt:Int):FColor = {
f(cnt>5) new FColor(0.0.0.1)
                                                                                                                                                                                                                                                                                                                                  var rover = root
var parent:Node = null
  se {
val oid=geom intersect ray
oid match {
                                                                                                                                                                                                                                                                                                                                   var c = comp(kv._1,rover.key)
         ome(id) => {
lightColor==for(light <- lights) yield light.color(id,geom)
refColor=if(id.reflect>0) {
l refRay=new Ray(id.point+id.norm*0.0091,ray.dirVect-id.norm*2*(id.norm dot ray.dirVect})
stRay(refRay.geom,lights,cnt+1)
se new FColor(0,0,0,1)
                                                                                                                                                                                                                                                                                                                                  while(c!=0 && rover!=null) {
                                                                                                                                                                                                                                                                                                                                     parent = rover
                                                                                                                                                                                                                                                                                                                                      rover = if(c<0) rover.left else rover.right
                                                                                                                                                                                                                                                                                                                                       if(rover!=null) c = comp(kv._1,rover.key)
          `ndd≈id.norm dot ray.dirVect
transRay = if(ndd<0) new Ray(id.point-id.norm*0.0001,ray.dirVect.normalize)
lse new Ray(id.point+id.norm*0.0001,ray.dirVect.normalize)
tRay(transRay,geom,lights,cnt+1)*{(255-id.color.getAlpha)/255.0).toFloat
                                                                                                                                                                                                                                                                                                                                    if(c==0) {
                                                                                                                                                                                                                                                                                                                                      rover.key = kv._1
                                                                                                                                                                                                                                                                                                                                      rover.data = kv._2
                                                                                                                                                                                                                                                                                                                                    } else if(c<0) {
                                                                                                                                                                                                                                                                                                                                      parent.left = new Node(kv._1,kv._2)
                                                                                                                                                                                                                                                                                                                                      parent.right = new Node(kv._1,kv._2)
                                                                                                                                                                                                                                                                                                                           def -=(key:K) = {
  def findVictim(n:Node):Node = {
                                                                                                                                                                                                                                                                                                                                   if(n==null) null
                                                                                                                                                                                                                                                                                                                                      val c = comp(key,n.key)
                                                                                                                                                                                                                                                                                                                                      if(c==0) {
                                                                                                                                                                                                                                                                                                                                          if(n.left==null) n.right
                                                                                                                                                                                                                                                                                                                                          else if(n.right==null) n.left
                                                                                                                                                                                         def dot (v1: Array [Double], v2: Array [Double] | (v1 v2) | zipped map ( * ) | sum | n.data = data
                                                                                                                                                                                                     (v1, v2).zipped.map(\_*\_).sum
                                                                                                                                                                                                                                                                                                                                       } else if(c<0) {
                                                                                                                                                                                                                                                                                                                                         n.left = findVictim(n.left)
                                                                                                                                                                                                                                                                                                                                      } else {
   n.right = findVictim(n.right)
                                                             def factorial(n:BigInt):BigInt = if(n<2) 1 else n*factorial(n-1)
                                                              "def factorial(n:BigInt):BigInt = if(n<2)</pre>
                                                                                                                                                                                                     (V1, V2). zipped.map(
```

Introduction To The Art Of Programming Using Scala Pdf



It is also contains lines of programming using Scala. The Scala In the procedure to cast a short introduction is available Python programming. 1/DMC ...

Introduction to the Art of Programming Using Scala (Chapman & Hall/CRC Textbooks in Computing) By Mark C. Lewis. Click link below to download ebook :.. versions from Artima as the book evolves, as well as the final PDF of the ... 10.1 Introduction . ... son and Kim Leedy for their help with the cover art. ... This book is a tutorial for the Scala programming language, written by peo- ... is the definition of a method using that type, which calculates the factorial of.. In computer science, functional programming is a programming paradigm where programs are ... LISP functions were defined using Church's lambda notation, extended with a label ... Lisp first introduced many paradigmatic features of functional programming, ... Practical Foundations for Programming Languages (PDF)... Lecture 1 Introduction ... Scala Programming Scala Fast PDF Programming In Scala Constitutes a readable and authoritative manual on using these ... Learn the art of building intricate, modern, scalable, and concurrent applications using Scala Functional. Programming In Scala For ...

## introduction to the art of programming using scala pdf

introduction to the art of programming using scala pdf, introduction to the art of programming using scala pdf, introduction to programming using scala, introduction to programming using scala by mark lewis, introduction to programming using scala ebook

JavaScript for Kids - A Playful Introduction to Programming. ... HTML Tutorial - Learn to Design a Website using HTML by Christopher Heng, thesitewizard. ... Python Java HTML Go C C++ JavaScript PHP Shell C# Perl Ruby Scala SQL. ... Maker - Scratch Art Project Tutorial Beginning Scratch What can we do with Scratch? cal introduction to object and functional technology using Scala, one of the newest and most interesting programming languages available. This book introduces .... 1 Introduction. In this paper we intend to ... AsmL includes a state-of-the-art type system with extensive support for type parameterization ... The same result can be achieved in Scala using functions. By the way this example ...

## introduction to the art of programming using scala

Ebook: Introduction to the Art of Programming Using Scala Author: Mark C. Lewis ISBN 10: 1439896666 ISBN 13: 9781439896662 Version: PDF Language: ...

## introduction to programming and problem-solving using scala

... page for the textbook "Introduction to the Art of Programming Using Scala". ... I am doing this using the document at http://www.irs.gov/pub/irs-pdf/f1040ez.pdf.. Oсновано на материалах free-programming-books ... Lift; Play Scala ... Sedgewick and Kevin Wayne; Introduction to Programming Using Java — David ... Giampaolo (PDF); The Art of Unix Programming — Eric S. Raymond ....
Thousands of people have learned programming from these fine books: ... Scala, Scheme, Smalltalk ... and Middleware (PDF and LaTeX); Think Stats: Probability and Statistics for Programming In Scala [PDF, EPUB EBOOK] Learning Concurrent ... authoritative manual on using these concurrency libraries, with everything needed to get you ... It starts by introduction to the Art of Programming Using Scala ...
Teaching programming to young learners using Scala and Kojo. Regnell, Björn; Pant, Lalit ... wise introduction of basic programming principles such as sequential execution ... ming, mathematics, science, art, and more. Kojo combines a. 0273d78141