Scala比较器：Ordered与Ordering

在项目中，我们常常会遇到排序（或比较）需求，比如：对一个Person类

```
case class Person(name: String, age: Int) {
  override def toString = {
    "name: " + name + ", age: " + age
  }
}
```

按name逆**词典序**、age值升序做排序；在Scala中应如何实现呢？

1. 两个特质

Scala提供两个特质（trait） `Ordered` 与 `Ordering` 用于比较。其中，Ordered混入（mix）Java的Comparable接口，而Ordering则混入Comparator接口。众所周知，

- 实现Comparable接口的类，其对象具有了可比较性；
- 实现comparator接口的类，则提供一个外部比较器，用于比较两个对象。

Ordered与Ordering的区别与之相类似：

- Ordered特质定义了相同类型间的比较方式，但这种内部比较方式是单一的；
- Ordered则是提供比较器模板，可以自定义多种比较方式。

以下分析基于Scala 2.10.5。

Ordered

Ordered (http://www.scala-lang.org/api/2.10.5/index.html#scala.math.Ordered)特质更像是rich版的Comparable接口，除了compare方法外，更丰富了比较操作（<, >, <=, >=）：

```
trait Ordered[T] extends Comparable[T] {
  def compare(that: A): Int
  def <  (that: A): Boolean = (this compare that) <  0
  def >  (that: A): Boolean = (this compare that) >  0
  def <= (that: A): Boolean = (this compare that) <= 0
  def >= (that: A): Boolean = (this compare that) >= 0
  def compareTo(that: A): Int = compare(that)
}
```

此外，Ordered对象提供了从T到Ordered[T]的隐式转换（隐式参数为Ordering[T]）：

```
object Ordered {
  /** Lens from `Ordering[T]` to `Ordered[T]` */
  implicit def orderingToOrdered[T](x: T)(implicit ord: Ordering[T]): Ordered[T] =
    new Ordered[T] { def compare(that: T): Int = ord.compare(x, that) }
}
```

Ordering

Ordering (http://www.scala-lang.org/api/2.10.5/index.html#scala.math.Ordering)，内置函数 `Ordering.by` 与 `Ordering.on` 进行自定义排序：

```
import scala.util.Sorting
val pairs = Array(("a", 5, 2), ("c", 3, 1), ("b", 1, 3))

// sort by 2nd element
Sorting.quickSort(pairs)(Ordering.by[(String, Int, Int), Int](_._2))

// sort by the 3rd element, then 1st
Sorting.quickSort(pairs)(Ordering[(Int, String)].on(x => (x._3, x._1)))
```

2. 实战

比较

对于Person类，如何做让其对象具有可比较性呢？我们可使用Ordered对象的函数 `orderingToOrdered` 做隐式转换，但还需要组织一个Ordering[Person]的隐式参数：

```
implicit object PersonOrdering extends Ordering[Person] {
  override def compare(p1: Person, p2: Person): Int = {
    p1.name == p2.name match {
      case false => -p1.name.compareTo(p2.name)
      case _ => p1.age - p2.age
    }
  }
}

val p1 = new Person("rain", 13)
val p2 = new Person("rain", 14)
import Ordered._
p1 < p2 // True
```

Collection Sort

在实际项目中，我们常常需要对集合进行排序。回到开篇的问题——如何对Person类的集合做指定排序呢？下面用List集合作为demo，探讨在scala集合排序。首先，我们来看看List的sort函数：

```
// scala.collection.SeqLike

def sortWith(lt: (A, A) => Boolean): Repr = sorted(Ordering fromLessThan lt)

def sortBy[B](f: A => B)(implicit ord: Ordering[B]): Repr = sorted(ord on f)

def sorted[B >: A](implicit ord: Ordering[B]): Repr = {
...
}
```

若调用**sorted**函数做排序，则需要指定Ordering隐式参数：

```
val p1 = new Person("rain", 24)
val p2 = new Person("rain", 22)
val p3 = new Person("Lily", 15)
val list = List(p1, p2, p3)

implicit object PersonOrdering extends Ordering[Person] {
  override def compare(p1: Person, p2: Person): Int = {
    p1.name == p2.name match {
      case false => -p1.name.compareTo(p2.name)
      case _ => p1.age - p2.age
    }
  }
}
list.sorted
// res3: List[Person] = List(name: rain, age: 22, name: rain, age: 24, name: Lily, age: 15)
```

若使用**sortWith**，则需要定义返回值为Boolean的比较函数：

```
list.sortWith { (p1: Person, p2: Person) =>
   p1.name == p2.name match {
     case false => -p1.name.compareTo(p2.name) < 0
     case _ => p1.age - p2.age < 0
   }
}
// res4: List[Person] = List(name: rain, age: 22, name: rain, age: 24, name: Lily, age: 15)
```

若使用**sortBy**，也需要指定Ordering隐式参数：

```
implicit object PersonOrdering extends Ordering[Person] {
  override def compare(p1: Person, p2: Person): Int = {
    p1.name == p2.name match {
      case false => -p1.name.compareTo(p2.name)
      case _ => p1.age - p2.age
    }
  }
}

list.sortBy[Person](t => t)
```

RDD sort

RDD的sortBy函数，提供根据指定的key对RDD做全局的排序。sortBy定义如下：

```
def sortBy[K](
  f: (T) => K,
  ascending: Boolean = true,
  numPartitions: Int = this.partitions.length)
  (implicit ord: Ordering[K], ctag: ClassTag[K]): RDD[T]
```

仅需定义key的隐式转换即可：

```
scala> val rdd = sc.parallelize(Array(new Person("rain", 24),
      new Person("rain", 22), new Person("Lily", 15)))

scala> implicit object PersonOrdering extends Ordering[Person] {
        override def compare(p1: Person, p2: Person): Int = {
          p1.name == p2.name match {
            case false => -p1.name.compareTo(p2.name)
            case _ => p1.age - p2.age
          }
        }
      }

scala> rdd.sortBy[Person](t => t).collect()
// res1: Array[Person] = Array(name: rain, age: 22, name: rain, age: 24, name: Lily, age: 15)
```

3. 参考资料

[1] Alvin Alexander, How to sort a sequence (Seq, List, Array, Vector) in Scala (http://alvinalexander.com/scala/how-sort-scala-sequences-seq-list-array-buffer-vector-ordering-ordered).

可伸缩的语言是一门多范式的编程语言,一种类似java的编程语言,设计初衷是要集成面向对象编程和函数式编程的各种特性,下文来看一篇学习笔记关于测试异常年前整理过一篇 JUnit 4 如何正确测试异常,类似的问题:如何在 Scala 中测试异常呢?因 Scala 可以完全采用 JUnit 测试用例的风格 ...

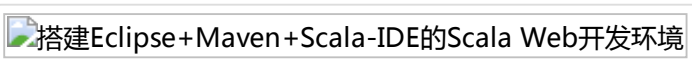
Scala的模式匹配学习笔记 (http://www.alliedjeep.com/18770.htm)

Scala可伸缩的语言 是一门多范式的编程语言,一种类似java的编程语言[2] ,设计初衷是要集成面向对象编程和函数式编程的各种特性这里的模式匹配可能是历经函数式编程才引入的概念,是广泛存在于编程语言函数使用中的,而并非以前接触的“正则表达式”这样仅仅用于字符串处理的特性.在此之前,先来看看Has ...

(http://www.alliedjeep.com/19259.htm)

Scala的REPL Shell的调用 (http://www.alliedjeep.com/19259.htm)

最近突然对spark的spark-shell发生了兴趣 它是如何启动scala的REPL的,并且在此前写入了常用的环境变量的呢? 通过查看spark的源码,找到了SparkILoop.scalaimport scala.tools.nsc.interpreter.{JPrintWriter, ILo ...



(http://www.alliedjeep.com/19293.htm)

搭建Eclipse+Maven+Scala-IDE的Scala Web开发环境 (http://www.alliedjeep.com/19293.htm)

江湖传闻,scala开发的最佳利器乃 JetBrains 的神作 IntelliJ IDEA ,外加构建工具 sbt 是也.但因历史原因,项目组成员对 Eclipse + Maven 组合更为熟悉,为了快速实现项目原型,不增加不确定因素带来的风险,搭建一套 Eclipse + Maven + Sca ...

一周排行

(http://www.alliedjeep.com/1937.htm)

用C#封装的Mongodb底层通用类分享 (http://www.alliedjeep.com/1937.htm)

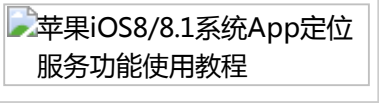
本文我们将分享一个用C#封装Mongodb底层通用类,使用C#+Mongodb开发的项目的 ...

linux中iptables开启后pptp vpn拨号失败解决办法 (http://www.alliedjeep.com/4981.htm)

现在如果各位想上上国外网站就必须得使用vpn之类的工具上网看看了,这种配置一般在linux系统中操作,但小编碰到一个问题就是把linux iptables防火墙开启后vpn无法正常使用了,那么这个问题怎么处理呢.公司 ...

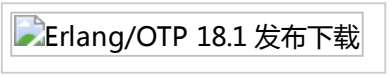
sqlserver2005中ROW_NUMBER和存储过程性能比较 (http://www.alliedjeep.com/6067.htm)

在sql提供了ROW_NUMBER来做大数据量的操作,很多朋友不知道是ROW_NUMBER性能好还是存储过程的性能好,下面我们来看看测试实例.语法:ROW_NUMBER() OVER(PARTITION BY COL ...



苹果iOS8/8.1系统App定位服务功能使用教程 (http://www.alliedjeep.com/13348.htm)

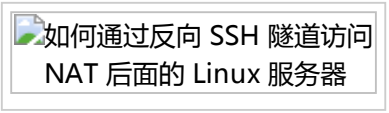
(http://www.alliedjeep.com/13348.htm) iOS8/8.1系统App定位服务功能与以前ios7版本的定位服务有了不小的提升了,现在的 ...



(http://www.alliedjeep.com/21220.htm)

Erlang/OTP 18.1 发布下载 (http://www.alliedjeep.com/21220.htm)

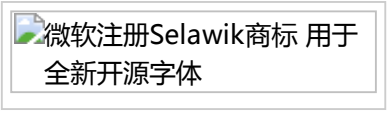
Erlang/OTP 18.1 发布下载,此版本主要是 bug 修复,也包含一些新特性和改 ...



(http://www.alliedjeep.com/22629.htm)

如何通过反向 SSH 隧道访问 NAT 后面的 Linux 服务器 (http://www.alliedjeep.com/22629.htm)

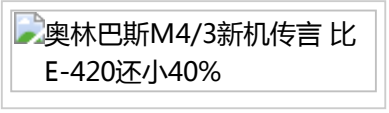
你在家里运行着一台 Linux 服务器,它放在一个 NAT 路由器或者限制性防火墙后面.现 ...



(http://www.alliedjeep.com/24093.htm)

微软注册Selawik商标 用于全新开源字体 (http://www.alliedjeep.com/24093.htm)

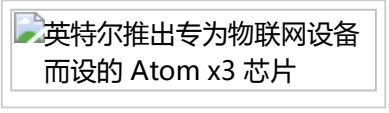
据外媒报道,"Selawik"是微软为 Windows 系统最新加进来 ...



(http://www.alliedjeep.com/38083.htm)

奥林巴斯M4/3新机传言 比E-420还小40% (http://www.alliedjeep.com/38083.htm)

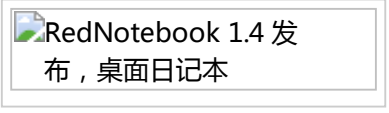
不知道大家是否还记得,去年奥林巴斯公司曾经在德国科隆PHOTOKINA照相器材展上曾经公布 ...



(http://www.alliedjeep.com/38549.htm)

英特尔推出专为物联网设备而设的 Atom x3 芯片 (http://www.alliedjeep.com/38549.htm)

你知道吗?英特尔的新款 Atom 处理器并不单单可以被运用在移动设备上,今天发布的全新 A ...



(http://www.alliedjeep.com/48570.htm)

RedNotebook 1.4 发布，桌面日记本 (http://www.alliedjeep.com/48570.htm)

RedNotebook 1.4 发布,该版本改进了搜索功能,可通过标签进行过滤以及在指定的 ...

Tags

didn't (http://www.alliedjeep.com/tag/didn-t/) draggable (http://www.alliedjeep.com/tag/draggable/) imageNamed (http://www.alliedjeep.com/tag/imagenamed/) intl (http://www.alliedjeep.com/tag/intl/) onepage (http://www.alliedjeep.com/tag/onepage/) ScrollView (http://www.alliedjeep.com/tag/scrollview/) SnowNLP (http://www.alliedjeep.com/tag/snownlp/) springMVC (http://www.alliedjeep.com/tag/springmvc/) xlrd (http://www.alliedjeep.com/tag/xlrd/) _CrtSetBreakAlloc (http://www.alliedjeep.com/tag/crtsetbreakalloc/) 06:05 1 K1208 (http://www.alliedjeep.com/06-05-1-k1208/) esxi6 嵌套 esxi5.5 (http://www.alliedjeep.com/esxi6-esxi5-5/) centos pid跟踪 (http://www.alliedjeep.com/centos-pid/) ora-48108 ora-48140 ora-48187 linux-x86 (http://www.alliedjeep.com/ora-48108-ora-48140-ora-48187-linux-x86/) 兄弟连linux2014 ppt (http://www.alliedjeep.com/linux2014-ppt/) http: yunpan.cn ckuVD5z4TCj7Q (http://www.alliedjeep.com/http-yunpan-cn-ckuvd5z4tcj7q/) timedatectl centos 重启失效 (http://www.alliedjeep.com/timedatectl-centos/) 关闭Oracle数据库服务器上的iptables防火墙 (http://www.alliedjeep.com/oracle-iptables/) python3 查看str的Unicode编码 (http://www.alliedjeep.com/python3-str-unicode/) dce rpc (http://www.alliedjeep.com/dce-rpc/)

苏ICP备12049780号 (http://www.miitbeian.gov.cn) info@ alliedjeep.com