Scala 数组 Scala List(列表)

Scala Collection

Scala提供了一套很好的集合实现，提供了一些集合类型的抽象。

Scala 集合分为可变的和不可变的集合。

可变集合可以在适当的地方被更新或扩展。这意味着你可以修改，添加，移除一个集合的元素。

而不可变集合类，相比之下，永远不会改变。不过，你仍然可以模拟添加，移除或更新操作。但是这些操作将在每一种情况下都返回一个新的集合，同时使原来的集合不发生改变。

接下来我们将为大家介绍几种常用集合类型的应用：

序号	集合及描述
1	Scala List(列表) List的特征是其元素以线性方式存储，集合中可以存放重复对象。 参考 API文档
2	Scala Set(集合) Set是最简单的一种集合。集合中的对象不按特定的方式排序，并且没有重复对象。 参考 API文档
3	Scala Map(映射) Map 是一种把键对象和值对象映射的集合，它的每一个元素都包含一对键对象和值对象。 参考 API文档
4	Scala 元组 元组是不同类型的值的集合
5	Scala Option Option[T] 表示有可能包含值的容器，也可能不包含值。
6	Scala Iterator（迭代器） 迭代器不是一个容器，更确切的说是逐一访问容器内元素的方法。

实例

以下代码判断，演示了所有以上集合类型的定义实例：

```
// 定义整型 List
val x = List(1,2,3,4)

// 定义 Set
val x = Set(1,3,5,7)
```

```
// 定义 Map
val x = Map("one" -> 1, "two" -> 2, "three" -> 3)

// 创建两个不同类型元素的元组
val x = (10, "Runoob")

// 定义 Option
val x:Option[Int] = Some(5)
```

Scala 数组 Scala List(列表)

1 篇笔记

写笔记

Scala 程序使用 Option 非常频繁，在 Java 中使用 null 来表示空值，代码中很多地方都要添加 null 关键字检测，不然很容易出现 NullPointException。因此 Java 程序需要关心那些变量可能是 null,而这些变量出现 null 的可能性很低，但一但出现，很难查出为什么出现 NullPointerException。

Scala 的 Option 类型可以避免这种情况，因此 Scala 应用推荐使用 Option 类型来代表一些可选值。使用 Option 类型，读者一眼就可以看出这种类型的值可能为 None。

参考链接：Scala 使用Option、Some、None，避免使用 null

starxhong 8个月前 (07-23)