Scala第十四章节

章节目标

1. 掌握隐式转换相关内容
2. 掌握隐式参数相关内容
3. 掌握获取列表元素平均值的案例

---

1. 隐式转换和隐式参数介绍

隐式转换和隐式参数是Scala中非常有特色的功能，也是Java等其他编程语言没有的功能。我们可以很方便地利用隐式转换来丰富现有类的功能。在后续编写Akka并发编程, Spark, Flink程序时都会经常用到它们。

- 隐式转换: 指的是用`implicit关键字`声明的带有`单个参数`的方法.
- 隐式参数: 指的是用`implicit关键字`修饰的变量.

> 注意: `implicit关键字`是在Scala的2.10版本出现的.

2. 隐式转换

2.1 概述

所谓**隐式转换**，是指`以implicit关键字声明的带有单个参数的方法`。该方法是**被自动调用**的，用来实现`自动将某种类型的数据转换为另外一种类型的数据`。

2.2 使用步骤

1. 在`object单例对象`中定义隐式转换方法.

   > 隐式转换方法解释: 就是用implicit关键字修饰的方法.

2. 在需要用到隐式转换的地方, 引入隐式转换.

   > 类似于`导包`, 通过`import关键字实现`.

3. 当需要用到`隐式转换方法`时, 程序会自动调用

2.3 示例一: 手动导入隐式转换方法

需求

通过隐式转换, 让File类的对象具备有read功能(即: 实现将文本中的内容以字符串形式读取出来).

步骤

1. 创建RichFile类，提供一个read方法，用于将文件内容读取为字符串
2. 定义一个隐式转换方法，将File隐式转换为RichFile对象
3. 创建一个File类的对象，导入隐式转换，调用File的read方法.

参考代码

```
import java.io.File
import scala.io.Source

//案例：演示 隐式转换，手动导入.
/*
  隐式转换:
    概述:
      用implicit修饰的 带有单个参数的方法，该方法会被自动调用.     //前提：需要手动引入.
    作用:
      用来丰富某些对象的功能的. 大白话解释：某个对象没有某个功能，通过特定手段让他具有此功能.
      //简单理解：这个类似于Java中的装饰设计模式.
      //BufferedReader br = new BufferedReader(new FileReader("a.txt))
      //这样写会报错，必须传入一个 要被升级功能的 对象.
      //BufferedReader br = new BufferedReader("a.txt")
 */
object ClassDemo01 {
  //1. 定义一个RichFile类，用来给普通的File对象添加 read()功能.
  class RichFile(file:File) {
    //定义一个read()方法，用来读取数据.
    def read() = Source.fromFile(file).mkString
  }

  //2. 定义一个单例对象，包含一个方法，该方法用于将：普通的File对象 转换成 RichFile对象.
  object ImplicitDemo {
    //定义一个方法，该方法用于将：普通的File对象 转换成 RichFile对象.
    implicit def file2RichFile(file:File) = new RichFile(file)
  }

  def main(args: Array[String]): Unit = {
    //3. 非常非常非常重要的地方：手动导入 隐式转换.
    import ImplicitDemo.file2RichFile

    //4. 创建普通的File对象，尝试调用其read()功能.
    val file = new File("./data/1.txt")
    /*
      执行流程:
        1. 先找File类有没有read()，有就用.
        2. 没有就去，查看有没有该类型的隐式转换，将该对象转成其他对象.
        3. 如果没有隐式转换，直接报错.
        4. 如果可以将该对象升级为其他对象，则查看升级后的对象中有没有指定方法，有，不报错，没有就报
错.
      如下的案例执行流程:
        1. file对象中没有read()方法.
        2. 检测到有 隐式转换将 file对象 转成 RichFile对象.
        3. 调用RichFile对象的read()方法，打印结果.
     */
    println(file.read())
  }
}
```

2.4 隐式转换的时机

既然 隐式转换 这么好用, 那什么时候程序才会 自动调用隐式转换方法呢?

1. 当对象调用类中不存在的方法或者成员时，编译器会自动对该对象进行隐式转换
2. 当方法中的参数类型与目标类型不一致时, 编译器也会自动调用隐式转换.

2.5 示例二: 自动导入隐式转换方法

在Scala中，如果 在当前作用域中有隐式转换方法 ，会自动导入隐式转换。

需求: 将隐式转换方法定义在main所在的 object单例对象 中

```
import java.io.File
import scala.io.Source

//演示 隐式转换，自动导入.
object ClassDemo02 {
  //1. 定义一个RichFile类，里边定义一个read()方法.
  class RichFile(file:File) {
    def read() = Source.fromFile(file).mkString
  }

  def main(args: Array[String]): Unit = {
    //2. 自定义一个方法，该方法用implicit修饰,
    //用来将: 普通的File -> RichFile，当程序需要使用的时候，会自动调用.
    implicit def file2RichFile(file:File) = new RichFile(file)

    //3. 创建File对象，调用read()方法.
    val file = new File("./data/2.txt")
    println(file.read())
  }
}
```

3. 隐式参数

在Scala的方法中, 可以带有一个 标记为implicit的参数列表 。调用该方法时, 此参数列表可以不用给初始化值, 因为 编译器会自动查找缺省值，提供给该方法 。

3.1 使用步骤

1. 在方法后面添加一个参数列表，参数使用implicit修饰
2. 在object中定义implicit修饰的隐式值
3. 调用方法，可以不传入implicit修饰的参数列表，编译器会自动查找缺省值

> 注意:
>
> 1. 和隐式转换一样，可以使用import手动导入隐式参数
> 2. 如果在当前作用域定义了隐式值，会自动进行导入

3.2 示例

需求

- 定义一个show方法，实现将传入的值，使用指定的前缀分隔符和后缀分隔符包裹起来.

> 例如: show("张三")("<<<", ">>>"), 则运行结果为: <<<张三>>>

- 使用隐式参数定义分隔符.
- 调用该方法，并打印结果.

参考代码

- 方式一: 手动导入隐式参数

```
//案例: 演示隐式参数，手动导入.
//演示参数: 如果方法的某个参数列表用implicit修饰了，则该参数列表就是: 隐式参数.
//好处: 我们再调用方法的时候，关于隐式参数是可以调用默认的值，不需要我们传入参数.
object ClassDemo03 {
  //需求: 定义一个方法，传入一个姓名，然后用指定的前缀和后缀将该名字包裹.
  //1. 定义一个方法show()，接收一个姓名，在接受一个前缀，后缀信息(这个是隐式参数).
  def show(name:String)(implicit delimit:(String, String)) = delimit._1 + name +
delimit._2

  //2. 定义一个单例对象，给隐式参数设置默认值.
  object ImplicitParam {
    implicit val delimit_defalut = "<<<" -> ">>>"
  }

  def main(args: Array[String]): Unit = {
    //3. 手动导入: 隐式参数.
    import ImplicitParam.delimit_defalut

    //4. 尝试调用show()方法.
    println(show("张三"))
    println(show("张三")("(((" -> ")))"))
  }
}
```

- 方式二: 自动导入隐式参数

```
//案例: 演示隐式参数，自动导入.
//演示参数: 如果方法的某个参数列表用implicit修饰了，则该参数列表就是: 隐式参数.
//好处: 我们再调用方法的时候，关于隐式参数是可以调用默认的值，不需要我们传入参数.
object ClassDemo04 {
  //需求: 定义一个方法，传入一个姓名，然后用指定的前缀和后缀将该名字包裹.

  //1. 定义一个方法show()，接收一个姓名，在接受一个前缀，后缀信息(这个是隐式参数).
  def show(name:String)(implicit delimit:(String, String)) = delimit._1 + name +
delimit._2


  def main(args: Array[String]): Unit = {
    //2. 自动导入 隐式参数.
    implicit val delimit_defalut = "<<<" -> ">>>"

    //3. 尝试调用show()方法.
    println(show("李四"))
    println(show("李四")("(((" -> ")))"))
```

```
    }
  }
```

4. 案例: 获取列表元素平均值

需求

通过隐式转换, 获取列表中所有元素的平均值.

目的

考察`隐式转换，列表`相关内容.

步骤

1. 定义一个RichList类, 用来给普通的List添加avg()方法，用于获取列表元素的平均值.
2. 定义avg()方法, 用来获取List列表中所有元素的平均值.
3. 定义隐式转换方法, 用来将普通List对象转换为RichList对象.
4. 定义List列表, 获取其中所有元素的平均值.

参考代码

```
object ClassDemo05 {
  //1. 定义一个RichList类，用来给普通的List添加avg()方法.
  class RichList(list:List[Int]) {
    //2. 定义avg()方法，用来获取List列表中所有元素的平均值.
    def avg() = {
      if(list.size == 0) None
      else Some(list.sum / list.size)
    }
  }

  //main方法，作为程序的主入口.
  def main(args: Array[String]): Unit = {
    //3. 定义隐式转换方法.
    implicit def list2RichList(list:List[Int]) = new RichList(list)

    //4. 定义List列表，获取其中所有元素的平均值.
    val list1 = List(1, 2, 5, 4, 3)
    println(list1.avg())
  }
}
```