

Scala第十五章节

章节目标

1. 了解递归的相关概述
2. 掌握阶乘案例
3. 掌握斐波那契数列案例
4. 掌握打印目录文件案例

1. 递归

递归指的就是 方法自己调用自己的情况。在涉及到复杂操作时，我们会经常用到它。在使用递归时，要注意以下三点：

1. 递归必须有出口，否则容易造成 死递归。
2. 递归必须要有规律。
3. 构造方法不能递归。
4. 递归方法必有 返回值的数据类型。

例如：下述的代码就是递归的写法。

```
def show() = {  
    show()  
}
```

2. 案例一：求阶乘

2.1 概述

所谓的阶乘其实指的是 数字1到该数字的累乘结果，例如5的阶乘就相当于 $5 * 4 * 3 * 2 * 1$ ，4的阶乘就相当于 $4 * 3 * 2 * 1$ ，根据上述的描述，我们可以得出两个结论：

1. 阶乘公式为(例如：求数字n的阶乘): $n! = n * (n - 1)!$
2. 1的阶乘等于1，即: $1! = 1$

2.2 需求

计算5的阶乘。

2.3 参考代码

```
//案例：求5的阶乘。
object ClassDemo01 {
    //1. 定义方法，用来求数字n的阶乘。
    def factorial(n:Int):Int = if(n == 1) 1 else n * factorial(n - 1)

    def main(args: Array[String]): Unit = {
        //2. 调用factorial方法，用来获取5的阶乘。
        val num = factorial(5)
        //3. 打印结果。
        println(num)
    }
}
```

2.4 内存图解

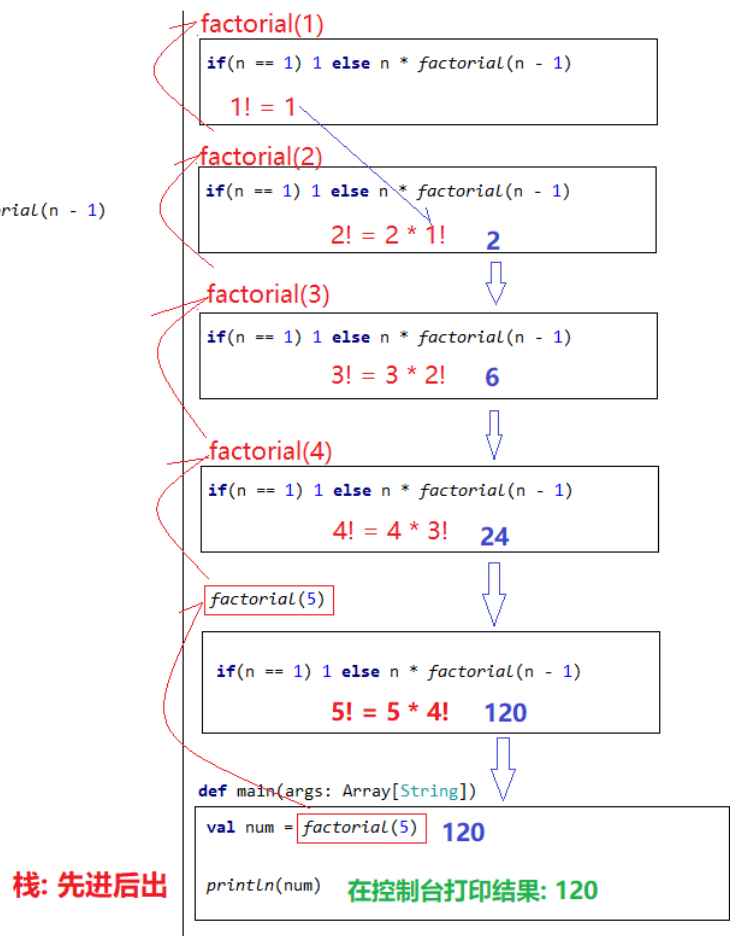
概述

在Scala中,内存被分为五部分,分别为 栈,堆,方法区,本地方法区,寄存器,特点如下:

- 栈
 - 功能:
 - 所有代码的执行.
 - 存储局部变量.
 - 特点: 按照 先进后出 的顺序执行,方法执行完毕后立马被回收.
- 堆:
 - 功能: 存储所有new出来的内容(即: 对象).
 - 特点: 在不确定的时间被 GC 回收.
- 方法区:
 - 功能: 存储字节码文件,方法等数据.
 - 特点: 程序执行完毕后,由操作系统来回收资源.
- 本地方法区:
 - 和 本地方法 相关,了解即可.
- 寄存器
 - 和 CPU 相关,了解即可.

阶乘图解

```
// 案例：求5的阶乘。  
object ClassDemo01 {  
  // 1. 定义方法，用来求数字n的阶乘。  
  def factorial(n: Int): Int = if(n == 1) 1 else n * factorial(n - 1)  
  
  def main(args: Array[String]): Unit = {  
    // 2. 调用factorial方法，用来获取5的阶乘。  
    val num = factorial(5)  
    // 3. 打印结果。  
    println(num)  
  }  
}
```



3. 案例二: 斐波那契数列

3.1 概述

传说在罗马时期有个意大利青年叫 斐波那契，有一天他提出了一个非常有意思的问题，假设：

1. 一对小兔子一个月之后会成长为一对大兔子。
2. 每一对大兔子每个月都会生一对小兔子。
3. 假设所有兔子都不死亡的情况下，问：1对小兔子，1年之后会变为多少对兔子？

3.2 思路分析

月份	兔子总对数	兔子详情
1	1	1对小兔子
2	1	1对大兔子
3	2	1对大兔子, 1对小兔子
4	3	2对大兔子, 1对小兔子
5	5	3对大兔子, 2对小兔子
12	?	?

即：已知数列1, 1, 2, 3, 5, 8, 13..., 问：第12个数字是多少？

3.3 参考代码

```
//案例：斐波那契数列
object ClassDemo02 {
    //1. 定义方法，用来获取兔子的对数.
    def rabbit(month: Int): Int = {
        if(month == 1 || month == 2) 1
        else rabbit(month - 1) + rabbit(month - 2)
    }

    def main(args: Array[String]): Unit = {
        //2. 调用方法，获取第12个月的兔子对数.
        val num = rabbit(12)
        //3. 打印结果.
        println(num)
    }
}
```

4. 案例三：打印目录文件

4.1 需求

1. 定义printFile(dir: File)方法, 该方法接收一个文件目录, 用来打印该目录下所有的文件路径.
2. 在main方法中测试printFile()方法.

4.2 目的

考察 递归, Java的File类 相关内容.

注意: 因为Scala是依赖VM的, 所以Java中的类库, Scala也可以无缝调用,

4.3 参考代码

```
import java.io.File

//案例：获取指定目录下所有的文件.
object ClassDemo03 {

    //1. 定义printFile()方法，用来打印指定目录下所有的文件信息.
    def printFile(dir: File): Unit = {
        if (!dir.exists()) {
            println("您录入的路径不存在")
        } else {
            val listFiles: Array[File] = dir.listFiles()
            for(listFile <- listFiles) {
                if(listFile.isFile) println(listFile)
                else printFile(listFile)
            }
        }
    }
}
```



```
}

//2.main方法，作为程序的主入口。
def main(args: Array[String]): Unit = {
    //3. 调用方法show()
    printFile(new File("d:\\abc"))
}
}
```