

第2章Scala基础实训

第2章Scala基础实训

CH02实训1：使用Scala编程实现杨辉三角

训练要点

需求说明

思路及步骤

作业要求

作业参考

在linux上编写scala文件

在linux上使用scalac编译

在linux上运行

预期结果

scala命令行运行：

学习视频

CH02实训2：使用Scala编写函数过滤文本中的回文单词

训练要点

需求说明

实现思路及步骤

作业要求

作业参考

预期结果：

学习视频

CH02实训3：使用Scala编程输出九九乘法表

训练要点

需求说明

实现思路及步骤

作业要求

作业参考

预期结果：

学习视频

GIT源码下载参考：

使用方法

GIT安装视频

CH02实训1：使用Scala编程实现杨辉三角

训练要点

- Scala函数的运用
- Scala循环的使用

需求说明

- 杨辉三角是二项式系数在三角形中的一种集合排列，在我国南宋数学家杨辉于1261年所著的《详解九章算法》一书里就已经记载了
- 基本关系：
1，每个数字都是它上方两数之和

- 2, 两侧最外围是由数字1组成
 - 3, 其余数则是其肩上两个数的和
- 请运用Scala编程实现指定层杨辉三角的输出

思路及步骤

1. 定义一个取值函数pascal,得到指定行列的值, 杨辉三角的性质决定了除左右两边等于1外, 元素的值等于其肩上两数的和, 即 $Value(row,column)=Value(row,column-1)+Value(row-1,column-1)$. 当行和列的值越来越小时达到终止条件
2. 以行为分界, 打印每个元素的值

作业要求

1. 使用Scala编写源码实现指定层杨辉三角的输出, 如:
vi ./yanghui.scala
2. 通过crt连接master主机,使用
scalac yanghui.scala 编译后, 运行:
scala -classpath . yanghui
3. 提交源码和脚本运行过程截图

作业参考

在linux上编写scala文件

```
1 | vi ./yanghui.scala
```

yanghui.scala内容

```
1 import scala.io.StdIn
2 object yanghui {
3   def main(args: Array[String]): Unit = {
4
5     val num = pascal()
6     val Value = Array.ofDim[Int](num,num)
7
8     for (i <- 0 until num) { //行
9       for (j <- 0 to i) { //列
10        if (j == 0 || j == i) value(i)(j) = 1
11        else value(i)(j) = value(i - 1)(j - 1) + value(i - 1)(j)
12        print(value(i)(j) + " ")
13      }
14      println()
15    }
16  }
17
18  def pascal():Int={
19    println("请输入行列值: ")
20    val num:Int = StdIn.readInt()
21    num
22  }
23 }
```

在linux上使用scalac编译

```
1 | scalac yanghui.scala
```

在linux上运行

```
1 | scala -classpath . yanghui
```

预期结果

```
1 | 请输入行列值:
2 | 11
3 | 1
4 | 1 1
5 | 1 2 1
6 | 1 3 3 1
7 | 1 4 6 4 1
8 | 1 5 10 10 5 1
9 | 1 6 15 20 15 6 1
10 | 1 7 21 35 35 21 7 1
11 | 1 8 28 56 70 56 28 8 1
12 | 1 9 36 84 126 126 84 36 9 1
13 | 1 10 45 120 210 252 210 120 45 10 1
```

scala命令行运行:

```
1 | val num = 10
2 | val value = Array.ofDim[Int](10,10)
3 |
4 | for (i <- 0 until num) { //行
5 |     for (j <- 0 to i) { //列
6 |         if (j == 0 || j == i) value(i)(j) = 1
7 |         else value(i)(j) = value(i - 1)(j - 1) + value(i - 1)(j)
8 |         print(value(i)(j) + " ")
9 |     }
10 |     println()
11 | }
```

学习视频

[学习视频](#)

CH02实训2: 使用Scala编写函数过滤文本中的回文单词

训练要点

1. 掌握Scala的数组、列表、映射的定义与使用。
2. 掌握Scala的for循环与if判断的使用。
3. 掌握Scala的函数式编程

需求说明

需求说明回文是指正向和逆向读起来相同的词，英语中也存在着回文现象，如"mom"和"dad"，英文中的回文单词给英语的学习增添了不少乐趣。现有一份英文文档word.txt,如表2—7所示，请使用Scala编程读取文件，并编写一个函数判断文档中的每个单词是否为回文单词，若是则输出该单词。

word.txt数据:

My father was my hero, all through my life. When I was very little, he appeared to be so large. In my eyes he could do anything, we all knew he was in charge. He was a man of great strength both physically and in mind, but in him there was a gentleness, he found ways to be outgoing and kind. Many days of childhood were greeted with a kiss, and songs to me as I awoke, those days I surely miss. He made me feel so special, Miss America he would sing. I knew I had my father's love. It gave me courage to do most anything.

实现思路及步骤

1. 读取word.txt数据，将数据放到缓存区。
2. 使用flatMap()方法获取缓存区里面的数据，并使用空格进行分割。
3. 定义函数isPalindrom(word: String)。
4. 在函数中判断单词正向与逆向是否一样，若是则输出该单词。
5. 调用isPalindrom函数

作业要求

提交源码和脚本运行过程截图

作业参考

```
1 package chap02
2
3 import scala.collection.mutable
4 import scala.io.Source
5
6 object Huiwen {
7   def main(args: Array[String]): Unit = {
8     // 读取数据
9     val src = Source.fromFile(请补充1)
10    val buffered = src.getLines().toBuffer
11    // 数据分割
12    val strings: mutable.Buffer[String] = buffered.flatMap(请补充2)
13    // 调用isPalindrom函数
14    for (word <- strings) {
15      isPalindrom(word)
16    }
17  }
```

```

18
19 // 定义isPalindrom(word: String)函数, 实现回文单词
20 def isPalindrom(word: String) = {
21
22     val word_reverse = word.reverse
23     if (请补充3) {
24         print(word + "\n")
25     }
26     else {
27         print("this word is not a palindrom" + "\n")
28     }
29 }
30 }
31

```

预期结果:

```

1 this word is not a palindrom
2 dad
3 this word is not a palindrom
4 this word is not a palindrom
5 .....

```

学习视频

[学习视频](#)

CH02实训3: 使用Scala编程输出九九乘法表

训练要点

1. 掌握Scala循环的使用
2. 掌握Scala函数式编程

需求说明

九乘法表是我国古代人民的智慧结晶, 在春秋战国时代就已在筹算中运算, 到明代则改良并用在算盘上。标准、规范的九九乘法表可以帮助学生更快地记住乘法口诀。现需要使用Scala编程输出九九乘法表, 要求输出的效果

```

1 * 1 = 1
1 * 2 = 2  2 * 2 = 4
1 * 3 = 3  2 * 3 = 6  3 * 3 = 9
1 * 4 = 4  2 * 4 = 8  3 * 4 = 12  4 * 4 = 16
1 * 5 = 5  2 * 5 = 10  3 * 5 = 15  4 * 5 = 20  5 * 5 = 25
1 * 6 = 6  2 * 6 = 12  3 * 6 = 18  4 * 6 = 24  5 * 6 = 30  6 * 6 = 36
1 * 7 = 7  2 * 7 = 14  3 * 7 = 21  4 * 7 = 28  5 * 7 = 35  6 * 7 = 42  7 * 7 = 49
1 * 8 = 8  2 * 8 = 16  3 * 8 = 24  4 * 8 = 32  5 * 8 = 40  6 * 8 = 48  7 * 8 = 56  8 * 8 = 64
1 * 9 = 9  2 * 9 = 18  3 * 9 = 27  4 * 9 = 36  5 * 9 = 45  6 * 9 = 54  7 * 9 = 63  8 * 9 = 72  9 * 9 = 81

```

实现思路及步骤

1. 定义函数fun(): Int)
2. 在fun函数中, 使用两个for循环构成算法结构
3. 调用函数fun(9) 输出九九乘法表

作业要求

提交源码和脚本运行过程截图

作业参考

```
1 package chap02
2
3 object jiujiu {
4     def main(args: Array[String]): Unit = {
5         fun(9)
6     }
7     def fun(n: Int) = {
8         for (请补充) {
9             for (请补充) {
10                printf("%d * %d = %d\t", j, i, j * i)
11            }
12            println()
13        }
14    }
15
16 }
17
```

预期结果:

```
1 1 * 1 = 1
2 1 * 2 = 2  2 * 2 = 4
3 1 * 3 = 3  2 * 3 = 6  3 * 3 = 9
4 1 * 4 = 4  2 * 4 = 8  3 * 4 = 12  4 * 4 = 16
5 1 * 5 = 5  2 * 5 = 10  3 * 5 = 15  4 * 5 = 20  5 * 5 = 25
6 1 * 6 = 6  2 * 6 = 12  3 * 6 = 18  4 * 6 = 24  5 * 6 = 30  6 * 6 = 36
7 1 * 7 = 7  2 * 7 = 14  3 * 7 = 21  4 * 7 = 28  5 * 7 = 35  6 * 7 = 42  7 * 7
  = 49
8 1 * 8 = 8  2 * 8 = 16  3 * 8 = 24  4 * 8 = 32  5 * 8 = 40  6 * 8 = 48  7 * 8
  = 56  8 * 8 = 64
9 1 * 9 = 9  2 * 9 = 18  3 * 9 = 27  4 * 9 = 36  5 * 9 = 45  6 * 9 = 54  7 * 9
  = 63  8 * 9 = 72  9 * 9 = 81
```

学习视频

[学习视频](#)

GIT源码下载参考:

使用方法

1. 安装Git-2.39.2-64-bit.exe和TortoiseGit-2.14.0.0-64bit.msi
2. 进入放置源码的路径，如D:spark
3. 右击空白，在菜单中选择git clone
4. 在url中粘贴：<https://jihulab.com/biglab-share/spark.git>
5. 确认，下载源码

GIT安装视频

[学习视频](#)