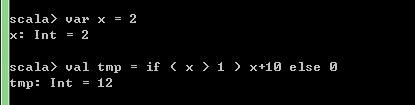
Scala 学习笔记之第二章-控制结构和函数

本章节学习scala使用条件表达式、循环和函数

要点：

1. If表达式有值，块也有值。
2. Scala的for循环是增强版的java 的for循环
3. 分号不必须
4. 函数定义中避免使用return
5. 函数定义中别漏掉 “=”
6. 异常内的catch中使用“模式匹配”
7. Scala没有受检异常
8. void 类型是unit（单元测试）
9. 条件表达式/(if表达式有值，块也有值)
   1. 将if表达式的值赋值给变量



Scala中的if/else 控制结构将java中的if/else结构和?:结构结合在了一起。

* 1. if/else内的表达式可以是不同类型

if ( x > 0 ) “YES ” else 0

两个表达式一个String类型，一个Int类型，但公共超类型为Any

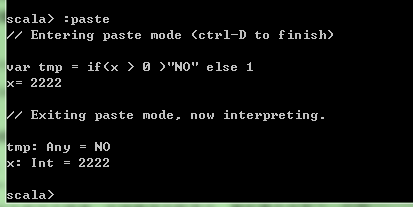
* 1. 有if结构

比如if(x > 0 ) 1 相当于 if(x > 0 ) 1 else ()

* 1. Scala没有switch语句

注意：

* + 1. REPL解析时只能看到一行代码，比如看到if但看不到else，但在被编译程序中没有这样的问题
    2. RPEL 模式下使用”:paste” 可以进入粘贴模式，最后按下Ctrl+D ,操作之间的代码会被当做整体分析的。



1. 语句终止

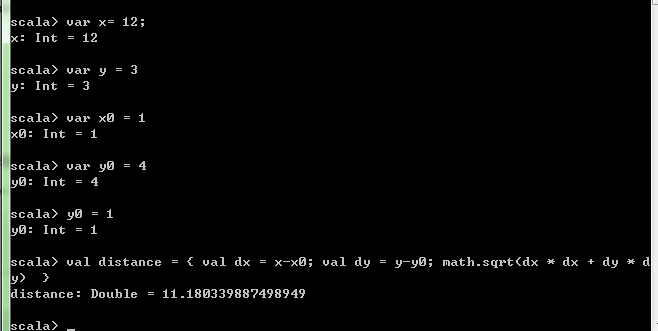
注意：

* + 1. 如果每行只有一条代码语句，不用写分号，如果多条语句写在同一行，则需要用分号隔开。
    2. 如果一条语句太长需要换行，以操作符结尾，避免解析器误判语句结束。

1. 块表达式和赋值

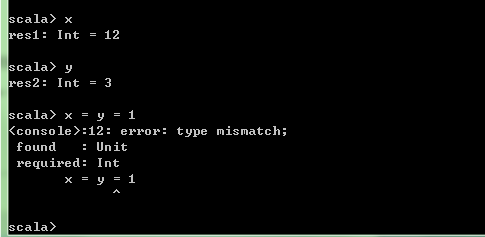
注意:

* + 1. Scala中，{}块包含一系列表达式，最后一个表达式的值就是块的值



最后计算的值就是distance的值。

* + 1. 以赋值语句结束的块的类型为Unit,因此语句不能串联在一起,会报错(需要的是Int类型，却找到Unit类型)。



1. 输入/输出

输入：

|  |  |  |  |
| --- | --- | --- | --- |
| readline 🡪 读一行 | readInt 🡪 读数字 | readDouble | readByte |
| readBoolean | readShort | readLong | readFloat |
| readChar |  |  |  |

注意：readLine需要输入提示信息

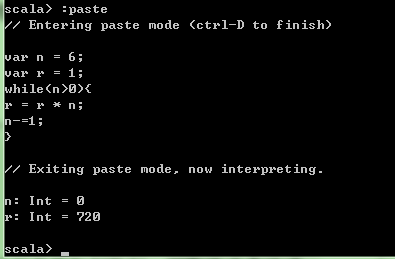
var定义的变量即使已经赋值，再次赋值时的类型也可以与原来不同。

输出：

|  |  |  |
| --- | --- | --- |
| print 🡪仅仅打印信息 | println 🡪 带有回车 | printf 🡪 C风格格式化 |

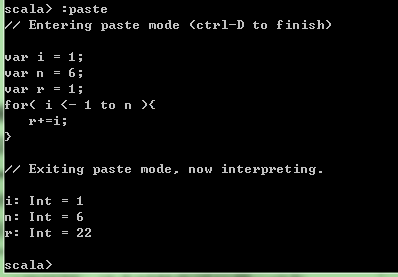
1. 循环

While：该循环和java中的相似



for:for(i <- n ) 让变量i遍历“<-“右边的表达式所有值。遍历具体如何执行，取决于表达式的类型。

在for循环的变量之前并没有val或者var的指定，该变量的类型是集合的元素类型。循环变量的作用域一直持续到循环结束。



For循环遍历字符串：

1. 算数和运算符重载