一、scala编程入门课
二、数据结构基础—排序

一、Scala是啥？

Scala是一门多范式的编程语言，一种类似java的编程语言，设计初衷是实现可伸缩的语言、并集成面向对象编程和函数式编程的各种特性。

Scala运行于Java平台（Java虚拟机），并兼容现有的Java程序。

二、数据类型

Scala共有8种数据类型，分别为：

7种数值类型：5种整数类型Byte、Short、Int、Long、char和两种浮点数类型Float、Double

1种布尔类型：Boolean

Scala的变量分两种：**val**和**var**。以val定义的变量名实际上是一个常量，例如：

```
Scala> val answer = 8
Answer: Int = 8
```

如果对它重新赋值，编译器则会报错，如：

```
Scala> answer = 4
<console>:6: error: reassignment to val
```

而要声明值可变的变量则使用var

三、开始你的scala编程

1.如何运行scala程序，（在命令行窗口）交互式
2.java –jar test.jar
3.在IDE中运行

四、流程控制

1. 条件表达式

scala的if/else语法结构和很多其他语言一样，首先测试某个条件，然后根据条件是否满足来执行两个不同代码分支当中的一个。例如:

```
if（x>0）1  else  -1
```

上述表达式值为1或-1，取决于x的值。可以将表达式的值赋给变量：

```
val s = if（x>0）  1  else  -1
```

Scala中，每个表达式都有一个类型，如表达式if（x>0）1 else -1的类型是Int。

如果else部分缺失了，那么if语句可能没有输出值。但是Scala中，每个表达式都应该有某种值。于是引入一个Unit类，写做（）。如：

```
if（x>0）1  else  ()
```

注意：赋值中，我们使用的是val而不是var。只要有机会，尽可能使用val，会让代码更容易读也更容易重构

四、流程控制

2. 循环语句——while循环

Scala拥有与Java和C++相同的while和do循环。包含了一个条件检查和一个循环体，只要条件检查为真，循环体就会一直执行，While循环如下：

```
While (n>0) {
        r = r*n
        n -=1
 }
```

注：while和do-while这样的语法结构，称之为“循环”而不是表达式，因为并不返回一个有意义的值

do-while是在循环体之后执行条件检查而不是在循环体之前，如：

```
do{
     r = r*n
     n -=1
} while(n>0)
```

注：Scala中并没有break语句或是continue语句来结束循环

四、流程控制

简单for语句，如下：

```
for(i<- 1 to n)
    r=r*i
```

其语法结构为for(i<-表达式)，让变量i遍历<-右边的表达式的所有值。

如遍历字符串或数组时，需要使用从0到n-1的区间，这时可以用until而不是to方法。

高级for语句：

可以以 ***变量<-表达式*** 的形式提供多个生成器，用分号隔开，如：

```
for(i<- 1 to 3;j<- 1 to 3)
    print ( (10*i+j)+" " )
//将打印11 12 13 21 22 23 31 32 33
```

四、流程控制

每个生成器都可以带一个守卫，以if开头的Boolean表达式：

```
for (i<- 1 to 3;j<- 1 to 3 if i!=j) print( (10*i+j)+" " )
        //将打印 12 13 21 23 31 32
```

可以使用任意多的定义，引入可以在循环中使用的变量：

```
for(i<- 1 to 3; from = 4 - i;j<- from to 3)
print( (10*i+j)+" " )
        //将打印13 22 23 31 32 33
```

如果for循环的循环体以yield开始，则该循环会构造出一个集合，每次迭代生成集合中的一个值：

```
    for (i<-1 to 10) yield i % 3
        //生成 Vector(1,2,0,1,2,0,1,0,1)
```

这类循环叫for推导式。

四、函数

4. 声明函数

函数定义中以def开始，给出函数名称、参数和函数体，如下：

```
def max(x: Int, y: Int): Int = {
    if (x > y)
        x
    else
        y
}
```

注：每个参数后必须加上以冒号开始的类标注

其中，max是函数名，x、y是参数，必须给出所有参数的类型如上例中的Int,其中第三个Int 是函数结果的类型，如果函数不是递归的则不需要指定返回类型。

四、函数

如果函数体需要多个表达式完成，可以用代码块。块中最后一个表达式的值就是函数的返回值，例如，下面这个函数返回位于for循环之后的r的值。

```
def fac (n: Int) = {
      var r = 1
for (i <- 1 to n) r = r*i
   r
 }
```

本例中，我们并不需要用到return。

四、函数

面向对象，使用类。 (课后自习)

四、常见工具类

数组

1)定长数组

如需一个长度不变的数组，可用scala中的Array。如：

```
val nums = new Array[Int](10)
    //10个整数的数组，所有元素初始化为0
val s = Array(“hello”,” world”)
    //已提供初始值，就不需要new
s(0) = “Goodbye”
    //使用（）而不是[ ]来访问元素
```

2)变长数组

长度按需要变化的数组，scala的数据结构为ArrayBuffer。如：

```
val b = ArrayBuffer[Int]()    //一个空数组缓冲
```

四、常见工具类

```
b +=1                          //用+=在尾端添加元素
b +=(1,2,3)                    //在尾端添加多个元素，以括号包起来
b ++=Array(8,9,10)             //用++=操作符追加任何集合
b.trimEnd(5)                   //移除最后5个元素
b.insert(2,6)                  //在下标2之前插入6
b.insert(2,4,5,6)              //在下标2之前插入4，5，6
b.remove(2)                    //删除下标为2的元素
b.remove(2,3)                  //从下标2开始移除3个元素
```

有时需要构建一个Array,但不知道最终需要装多少元素，这种情况，先构建一个数组缓冲，然后调用：

```
b.toArray
//Array(1,1,2)
```

四、常见工具类

HashMap

```
val map = new HashMap[String, Int]()
  map.put("李二", 100)
  map.put("Tom", 77)
  println(map.get("Tom").get)
  map.remove("Tom")
  println(map)
```

四、常见工具类

List 和 元组

```
    val list = new ListBuffer[Int]()
    list.append(3)
    list.append(2)
    list.appendAll(Array(4,5))
    println(list)
    println(list(3))
    println(list.remove(0))
    println(list)

    val tuple = ("AA", 1, "CC")
    println(tuple._1)
//    tuple(0) = 3
```

五、处理异常

```
try{
   val f = new FileReader(“input.txt”)
} catch{
  case ex:FileNotFoundException => // 处理找不到文件的情况
  case ex: IOException => //处理其他I/O错误
```

try-catch表达式首先代码体会被执行，如果抛出异常，则依次尝试每个catch子句。

try/finally语句可以释放资源，不论有没有异常发生。

```
try{
process(file)
}  finally{
file.close()     //确保关闭文件
 }
```

finally 语句不论process函数是否抛出异常都会执行，reader总会被关闭。

五、处理异常

Scala 的异常处理跟其他语言类似。当需要主动抛出异常时，如：

```
throw new IllegalArgumentException(“ n must be even”)
```

当前运行被终止，运行时系统查找可以接受IllegalArgumentException的异常处理器。抛出的对象必须是java.lang.Throwable的子类。

throw表达式有特殊的类型Nothing。这在if/else表达式中很有用，如果一个分支的类型是Nothing,则if/else表达式的类型就是另一个分支的类型。

课堂练习:

1.开发一个通讯录
2.模拟数据雨