1. scala中的闭包？

闭包的实质就是代码与用到的非局部变量的混合。

说白了就是在函数体内可以捕获作用域外的变量

val y = 5

Val sum = (x: Int)=> x + y

Sum(5)

结果是10，说明捕获了y

1. scala中的柯理化？

柯里化(Currying)指的是将原来接受两个参数的函数变成新的接受一个参数的函数的过程。新的函数返回一个以原有第二个参数为参数的函数。

def mul(x:Int,y:Int) = x \* y //该函数接受两个参数

def mulOneAtTime(x:Int) = (y:Int) => x \* y //该函数接受一个参数生成另外一个接受单个参数的函数

这样的话，如果需要计算两个数的乘积的话只需要调用：

mulOneAtTime(5)(4)

1. scala中的case class与class的区别？

Case class是样本类具有以下特点：

①自动添加与类名一样的构造函数（也就是伴生对象，通过apply实现）也就是说构造对象的时候不需要new

②样本类中参数默认是val的，也就是说是privite final修饰的，不可以更改

③默认实现了toString、equals、hashCode、copy方法

④样本类可以通过==来比较两个对象

Class是一个类，构造对象需要new

1. 谈一谈scala中的隐式转换？

所谓的隐式转换函数(implicit conversion function)指的事那种以implicit关键字申明的带有单个参数的函数，这样的函数会被自动的应用，将值从一种类型转换为另一种类型。大多应用在增强功能情况下，如File对象本身没有read()函数

//这里的RichFile相当于File的增强类 需要将被增强的类作为参数传入构造器中

class RichFile(val file: File) {

def read = {

Source.fromFile(file.getPath).mkString

}

}

//implicit是隐式转换的关键字 这里定义一个隐式转换函数把当前类型转换成增强的类型

object Context {

//File --> RichFile

implicit def file2RichFile(file: File) = new RichFile(file)

}

//导入隐式转换

import Context.file2RichFile

//File类本身没有read方法 通过隐式转换完成

//这里的read方法是RichFile类中的方法 需要通过隐式转换File --> RichFile

println(new File("E:\\projectTest\\1.txt").read)

1. scala中伴生类和伴生对象是怎么回事？

在scala中，单例对象与类同名时，该对象被称为伴生对象，而这个类称为这个单例对象的伴生类。伴生类和伴生对象要在同一个源文件中定义，伴生对象和伴生类可以相互访问其私有成员。