**Scala 之Apply**

https://blog.csdn.net/shenlei19911210/article/details/78538255

将对象以函数的方式进行调用时，scala会隐式地将调用改为在该对象上调用apply方法。

https://blog.csdn.net/pzw\_0612/article/details/48576569

Scala比Java更面向对象的一个方面是Scala没有静态成员。替代品是，Scala有单例对象：singleton object。

当单例对象与某个类共享同一个名称时，他被称作是这个类的伴生对象：companion object。你必须在同一个源文件里定义类和它的伴生对象。类被称为是这个单例对象的伴生类：companion class。类和它的伴生对象可以互相访问其私有成员。

定义单例对象不是定义类型（在Scala的抽象层次上说）

类和单例对象间的一个差别是，单例对象不带参数，而类可以。因为你不能用new关键字实例化一个单例对象，你没机会传递给它参数。每个单例对象都被作为由一个静态变量指向的虚构类：synthetic class的一个实例来实现，因此它们与Java静态类有着相同的初始化语法。Scala程序特别要指出的是，单例对象会在第一次被访问的时候初始化。

Scala 的apply 有2 张形式，一种是 伴生对象的apply ，一种是 伴生类中的apply，下面展示这2中的apply的使用。

示例代码：

class ApplyOperation {

}

class ApplyTest{

def apply() = println("I am into spark so much!!!")

def haveATry: Unit ={

println("have a try on apply")

}

}

object ApplyTest{

def apply() = {

println("I am into Scala so much")

new ApplyTest

}

}

object ApplyOperation{

def main (args: Array[String]) {

val array= Array(1,2,3,4)

val a = ApplyTest() //这里就是使用object 的使用

a.haveATry

a() // 这里就是 class 中 apply使用

}

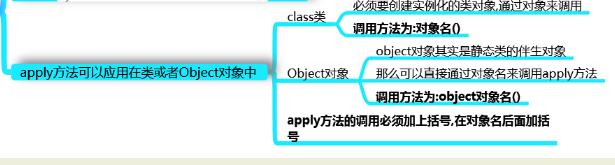
}

运行结果

I am into Scala so much

have a try on apply

I am into spark so much!!!



object apply 是一种比较普遍用法。 主要用来解决复杂对象的初始化问题。同时也是单例