

## Scala class和case class的区别

在Scala中存在case

class，它其实就是一个普通的class。但是它又和普通的class略有区别，如下：

1、初始化的时候可以不用new，当然你也可以加上，普通类一定需要加new；

```
scala> case class Iteblog(name:String)
defined class Iteblog
```

```
scala> val iteblog = Iteblog("iteblog_hadoop")
iteblog: Iteblog = Iteblog(iteblog_hadoop)
```

```
scala> val iteblog = new Iteblog("iteblog_hadoop")
iteblog: Iteblog = Iteblog(iteblog_hadoop)
```

2、toString的实现更漂亮；

```
scala> iteblog
res5: Iteblog = Iteblog(iteblog_hadoop)
```

3、默认实现了equals 和hashCode ；

```
scala> val iteblog2 = Iteblog("iteblog_hadoop")
iteblog2: Iteblog = Iteblog(iteblog_hadoop)
```

```
scala> iteblog == iteblog2
res6: Boolean = true
```

```
scala> iteblog.hashCode
res7: Int = 57880342
```

4、默认是可以序列化的，也就是实现了Serializable ；

```
scala> class A
```

defined class A

```
scala> import java.io._  
import java.io._
```

```
scala> val bos = new ByteArrayOutputStream  
bos: java.io.ByteArrayOutputStream =
```

```
scala> val oos = new ObjectOutputStream(bos)  
oos: java.io.ObjectOutputStream = java.io.ObjectOutputStream@4c257aef
```

```
scala> oos.writeObject(iteblog)
```

```
scala> val a = new A  
a: A = $iwC$$iwC$A@71687b10
```

```
scala> oos.writeObject(a)  
java.io.NotSerializableException: $iwC$$iwC$A
```

5、自动从scala.Product中继承一些函数;

6、case class构造函数的参数是public级别的，我们可以直接访问；

```
scala> iteblog.name  
res11: String = iteblog_hadoop
```

7、支持模式匹配；

其实感觉case class最重要的特性应该就是支持模式匹配。这也是我们定义case class的唯一理由，难怪Scala官方也说：It makes only sense to define case classes if pattern matching is used to decompose data structures.。来看下面的例子：

```
object TermTest extends scala.App {  
  def printTerm(term: Term) {  
    term match {  
      case Var(n) =>  
        print(n)  
      case Fun(x, b) =>  
        print("^" + x + ".")  
        printTerm(b)  
      case App(f, v) =>  
        print("(")
```

```
    printTerm(f)
    print(" ")
    printTerm(v)
    print("")
  }
}
def isIdentityFun(term: Term): Boolean = term match {
  case Fun(x, Var(y)) if x == y => true
  case _ => false
}
val id = Fun("x", Var("x"))
val t = Fun("x", Fun("y", App(Var("x"), Var("y"))))
printTerm(t)
println
println(isIdentityFun(id))
println(isIdentityFun(t))
}
```

本博客文章除特别声明，全部都是原创！

原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。

本文链接: [【】](#) ( )