# 课堂笔记

**object、class、trait是Scala里最核心概念。**

**在Java中：**

**var** p = **new** People() 中p称作实例化对象，或实例.

scala里也一样，注意实例化对象和object不是一回事。

类class里无static类型，类里的属性和方法,必须通过new出来的对象，所以有main主函数也没用。

而object的特点是：

1、可以拥有属性和方法，且默认都是“static”类型，可以用object名直接调用属性和方法，不需要通过new出来的对象（也不支持）。

2、object里的main函数是应用程序的入口。

object还有很多和class相同的地方，可以extends父类或Trait（Scala里接口的概念）。

但object不可以extends object，即object无法作为父类。

Java里的源码文件名称必须要以文件内的class为名，但是scala却不需要。

不过根据习惯通常用class名为scala文件名，便于查找。

Trait 相当于Java中的Interface，但有诸多不同，后面章节详细讲解。

## 类

关键词class

**class** DBOperate() //()可以去掉

{

var a = 10

**def** insert(table:*String*, id:Int, name :*String*) =

{

println("insert into "+table +","+id+","+name)

}

**def** success() =

{

println("success ")

}

}

类里面函数（方法），常量和变量称作类的属性

调用：

**val** dbOper:DBOperate = **new** DBOperate()

dbOper.insert("Test", 1, "name")

dbOper.success()

dbOper.success //无参数时，可以不写（）

### 构造函数

一个主构造器（函数），其他是辅助构造器

辅助构造器的实现体里，必须引用（调用）主构造器

主构造器的参数，也会成为类的属性？

//Person 类

辅助构造函数的名称都是this

辅助构造函数中必须以一个其他辅助构造器或主构造器的调用开始

class的默认主构造函数是无参数，比如Person()

**class** Person()

{

**private** **var** name = ""

**private** **var** age = 0

**def** **this**(name :*String*)

{

**this**() //调主构造器

**this**.name = name

}

**def** **this**(name :*String*, age:Int)

{

**this**(name)

**this**.age = age

}

}

调用：

**var** p1 = **new** Person() //主构造器

**var** p2 = **new** Person("zhangsan")

**var** p3 = **new** Person("zhangsan",20)

当主构造器中的不带var或val的参数，被至少1个方法使用时，这些参数会升格为字段，比如：

class Person(name :String, age:Int)

{

def func = name + “ is ” + age “ years old”

}

### Bean 属性

当字段被标为@BeanProperty 时，例如：

**import** scala.beans.BeanProperty

**class** Person2

{

@BeanProperty **var** name : *String* = "sdfg"

// 有了@BeanProperty，出现get和set方法，可以用get或set方法访问之

// 没有@ BeanProperty修饰的，可以直接访问之，但是没有get和set方法

}

调用：

**var** p4 = **new** Person2

println(p4.getName())

### 构造函数

scala类名称后直接加上构造函数的参数列表，这个构造函数就是主构造函数。另外，Scala中有只有一个主要构造函数，其他都是辅助构造函数。而且需要注意的是，辅助构造函数必须调用主构造函数或者通过this(…)之间相互调用。

### 访问修饰符

包、类、对象的属性和方法可以标记访问修饰符private和protected，如果我们不使用这两种关键字，那么访问将被默认设置为public。

访问修饰符是对包、类、object的属性和方法 访问权限的限制。

public 的成员可以从任何地方访问，默认

private 类型，必须同个.scala类文件内部访问。

protected的成员是从该成员定义的类自身和其子类才能访问，很少用。