# 课堂笔记

思考一下，为什么不可以继承多个父类？

原因是当多个父类里有相同函数或属性时，无法控制用哪个。

## Trait 特质

和Java里的接口(Interface)最相似，但又不同，最大便利就是**代码复用**。

trait不可以被new实例。

Scala和Java一样不允许子类extends多个父类。所以对class来讲代码复用度上受限。

Java里用Interface（接口）来解决这个问题，接口里定义公共函数（不可以有函数体实现），类可以implements多个接口。

Scala的Trait相当于Java里的Interface，但Trait不仅可以定义函数，还可以有函数体实现。类可以实现任意多个Trait。

实现关键词还是extends, 实现多个Trait用with，如：

final class AnyRefMap **extends** AbstractMap with Map with MapLike

当extends的多个Trait里有相同函数时，子类必须重写该函数。

规则：

1. 父trait里无函数体的函数，子类必须override。
2. 重写父类里有函数体的函数，必须有关键词**override。**
3. **trait里的变量，都是val类型。**
4. **在trait里定义的变量，必须是val类型，如果变量没初始化，子类必须override。**

### 同时extends类和Trait时

**class** OrderDAOImpl **extends** Father **with** OrderDAOTrait **with** BaseTrait {

其中Father 为class

OrderDAOTrait 和 BaseTrait 为Trait

其中OrderDAOTrait 和 BaseTrait的顺序不限制，Father 必须紧跟extends关键字。

### JVM如何编译Trait

案例1：

**trait** TestTrait {

**def** fun()

}

反编译后是什么？

public abstract class com.les2.TestTrait$class {

public static void fun(com.les2.TestTrait);

public static void $init$(com.les2.TestTrait);

}

当**def** fun()改为**def** fun(){} 时，反编译结果如何？

public abstract class com.les2.TestTrait$class {

public static void fun(com.les2.TestTrait);

public static void $init$(com.les2.TestTrait);

}

案例2：

**trait** TestTrait {

**def** fun(){

println("---")

}

**def** fun200()

}

编译后是什么？

统一编程规范！

掌握了（集合类）List的操作，就会Set、Seq、Range、Array

也就掌握了Spark的RDD（集合类）操作。

Spark Core RDD开发；

Spark DataFrame ;

Spark Streaming;