第1节 A two-dimensional layout library

第2节 Abstract classes 抽象类

1.创建抽象类

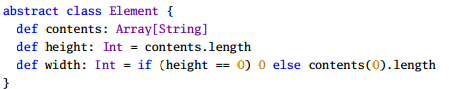
abstract class Element { def contents: Array[String] }

使用abstract描述一个class,该class的方法可以不用实现

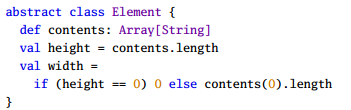
因此该class也不能直接new

第3节 Defining parameterless methods 定义无参数的方法

1.定义两个无参数方法,因此可以将参数()取消



2.无参数的方法可以转换成属性,将def转换成val



方法和属性的区别是,属性是占用内存的,初始化的时候就会把属性初始化,然后存储在内存中,而方法是每次执行调用的时候都会调用一次。

即属性只是在初始化的时候调用一次,不会每次调用属性都去初始化,但是方法是每次调用都会被调用方法的逻辑。

具体参见第6章8节的demo,class的执行逻辑。

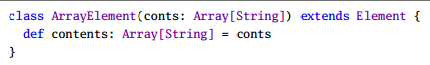
3.执行的时候,也因为没有参数,可以取消()



但是不建议,因为这样就没办法区分是方法还是属性了,因此还是调用方法的时候加入()是对的。

第4节 Extending classes 对类的扩展,使用extends继承

1.继承语法



2.注意事项

a.arrayElement是Element的子类

b. arrayElement继承了所有Element中非私有的成员

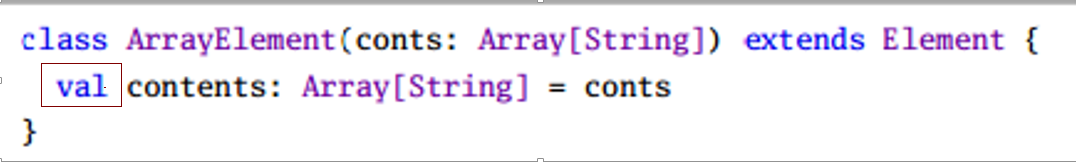
c.子类中对父类相同的name以及参数的也是会覆盖的,不会被继承

3.虽然是子类,但是依然可以当作父类来使用

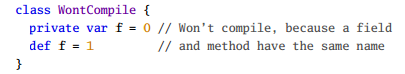
val e: Element = new ArrayElement(Array("hello"))

第5节 Overriding methods and fields

1. scala的field和method属于相同的命名空间,因此一个field可以覆盖一个无参数的方法是可行的。



2.因为scala在一个class中,对于field和method是不允许有相同名字的,但是java是可以的。

在scala中是有问题的

3.scala使用val去override覆盖无参数的method原理

scala有两个命名空间,而java有4个

a.java的命名空间fields, methods, types, and packages.

b.scala的命名空间

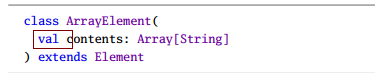
• values (fields, methods, packages, and singleton objects)

• types (class and trait names)

因此scala的methods和fields在相同的命名空间下,因此我们可以使用val去override覆盖无参数的method

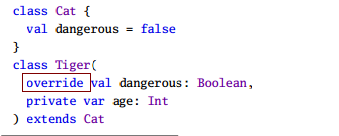
第6节 Defining parametric fields 定义参数field去实现父类

1.因为上面的例子,参数又被copy了一份,代码冗余,而且味道不好,因此使用参数名字与父类相同的方式,可以优化代码

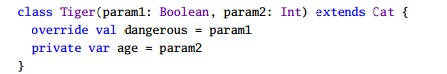


Val指示该参数不会被修改

2.或者使用任意参数名字,但是子类可以使用override

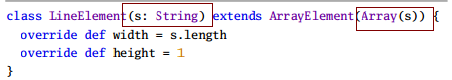


或者



第7节 Invoking superclass constructors 执行父类的构造函数

1.通过()方式为父类的构造函数添加真实的数据



第8节 Using override modifiers

1.规则

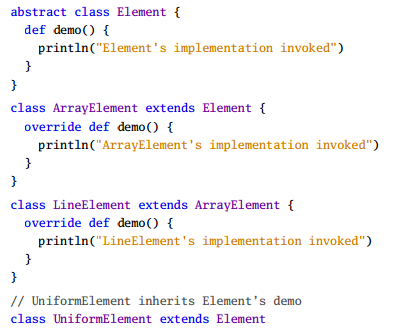
如果实现了父类的abstract成员,则override是可以选择不写的

如果子类没有覆盖或者实现父类的方法和属性的时候,则override关键字是不允许存在的

如果去覆盖父类已经存在的具体的方法,则必须要写override,这样主要的目的是让代码有更详细的信息

第9节 Polymorphism and dynamic binding 多态和动态绑定

1．定义一个抽象类以及多个实现类



2. 定义一个方法,接收参数是抽象类,具体的是什么,取决于动态传递进来的是什么类型



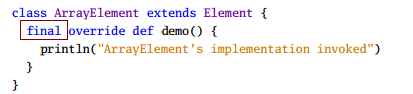
3．调用



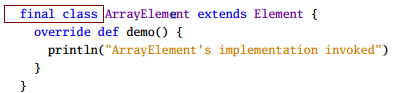
第10节 Declaring final members 声明final的成员变量

1.有时候我们定义一个成员的时候,不想被子类overridden,因此跟java一样,添加final修饰符即可

2.声明一个方法不能被覆写



3. 声明一个class不能有子类



第11章 Using composition and inheritance 使用组合和继承

暂时觉得没什么意义也没看懂

第12章Implementing above, beside, and toString

剩下的不太知道他在做什么