关于java的访问修饰符的问题：

（1）

Java的访问修饰符就是去规定类以及类中的属性和方法的访问权限，java中一共有四种访问修饰符 public private protected default。

其中default比较特殊：default不能修饰属性，default只能修饰接口中的方法，不能修饰普通类中的方法

只有public可以修饰类

（2）

private，私有的访问权限，也是最严格的访问权限，仅只能在设置了该权限的类中访问，利用这个访问权限，表现出封装思想。

default，默认的访问权限，也是可以省略的访问权限，它不仅能在设置了该权限的类中访问，也可以在同一包中的类或子类中访问。

protected，受保护的访问权限，它除了具有default的访问权限外，还可以在不同包中所继承的子类访问。

public，公有的访问权限，也是最宽松的访问权限，不仅可以是同一个类或子类，还是同一个包中的类或子类，又还是不同包中的类或子类，都可以访问。

-------------------------------------------------------------------------------------------------------------------------------------------------------------------

java8之前接口是不允许有方法体的，java8之后允许static/default 关键字修饰的方法可以有方法体，并且必须有方法体 ! ! !子类在实现含有static/default 关键字修饰的方法的接口时，可以不重写这些方法，也可以重写（这时接口中方法会被子类重写方法覆盖）（但是接口中的stati方法是不会被子类继承的，因此谈不上重写）

(1）

default关键字可以让接口中的方法可以有默认的函数体，当一个类实现这个接口时，可以不用去实现这个方法，当然，这个类若实现这个方法，就等于子类覆盖了这个方法，最终运行结果符合Java多态特性。（default关键字定义的方法必须有方法体，实现类就默认拥有该方法，当然这个默认的方法也可以重写。）

//那么这种新特性有什么意义呢？

假设实现这个接口的类有100个，并且这100个类中都会对某一个接口方法有同样的实现，那么只要在父类将方法体实现即可，不必在这100个类中挨个实现，看下面这个例子你就理解接口方法有方法体的意义了。

interface Formula {

double calculate(int a);

default double sqrt(int a) { return Math.sqrt(a); }

}

Formula接口在拥有calculate方法之外同时还定义了sqrt方法，实现了Formula接口的子类只需要实现一个calculate方法，默认方法sqrt将在子类上可以直接使用。

public class FormulaImpl impleaments Formula {

public static void main(String[] args) {

Formula formula = new Formula() {

@Override

public double calculate(int a) {

return sqrt(a \* 100);

}

};

System.out.println(formula.calculate(100));

// 100.0 System.out.println(formula.sqrt(16)); // 4.0 } }

（2）

/\*\*

 \*  可以使用default关键字,static关键字定义接口

 \* Java8新特性:JDK1.8以前,要定义一个接口,接口中只能定义全局常量和抽象方法

 \* Java8新特性:JDK1.8以后,接口的定义被打破了,要定义一个接口,接口中不仅可以定义全局常量和抽象方法

 \* 接口中还可以使用default关键字定义普通方法,这个方法有方法体,而且子类默认继承,不需要必须复写此方法,通过类的实例化对象来调用此方法

 \* 接口中还可以使用static关键字定义静态/类方法,这个方法有方法体,而且子类继承不了接口中的static方法,因此更谈不上子类复写此方法,只能通过接口的名字来调用此方法

 \*

 \*/

（3）

1.8版本是允许implements多个接口，但是会在编译阶段报错要求开发者必须override同名的方法

最后，我们有一点需要注意的地方，如果一个类实现了多个接口，恰巧多个接口中有相同的default修饰的方法，那么这个类必要要手动实现该默认方法，否则会报错。因为类的实例化对象不知道该调用哪个父类的efault修饰的方法，所以必须子类重写多个接口中有相同的default修饰的方法

（4）

注意，实现接口的类或者子接口不会继承接口中的静态方法，这个静态方法的只能由接口名调用，不能有子类对象调用

如果接口中的默认方法不能满足某个实现类需要，那么实现类可以覆盖默认方法。

————————————————

版权声明：本文为CSDN博主「Mar.三月」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/strivenoend/article/details/83044371