Java 面向对象

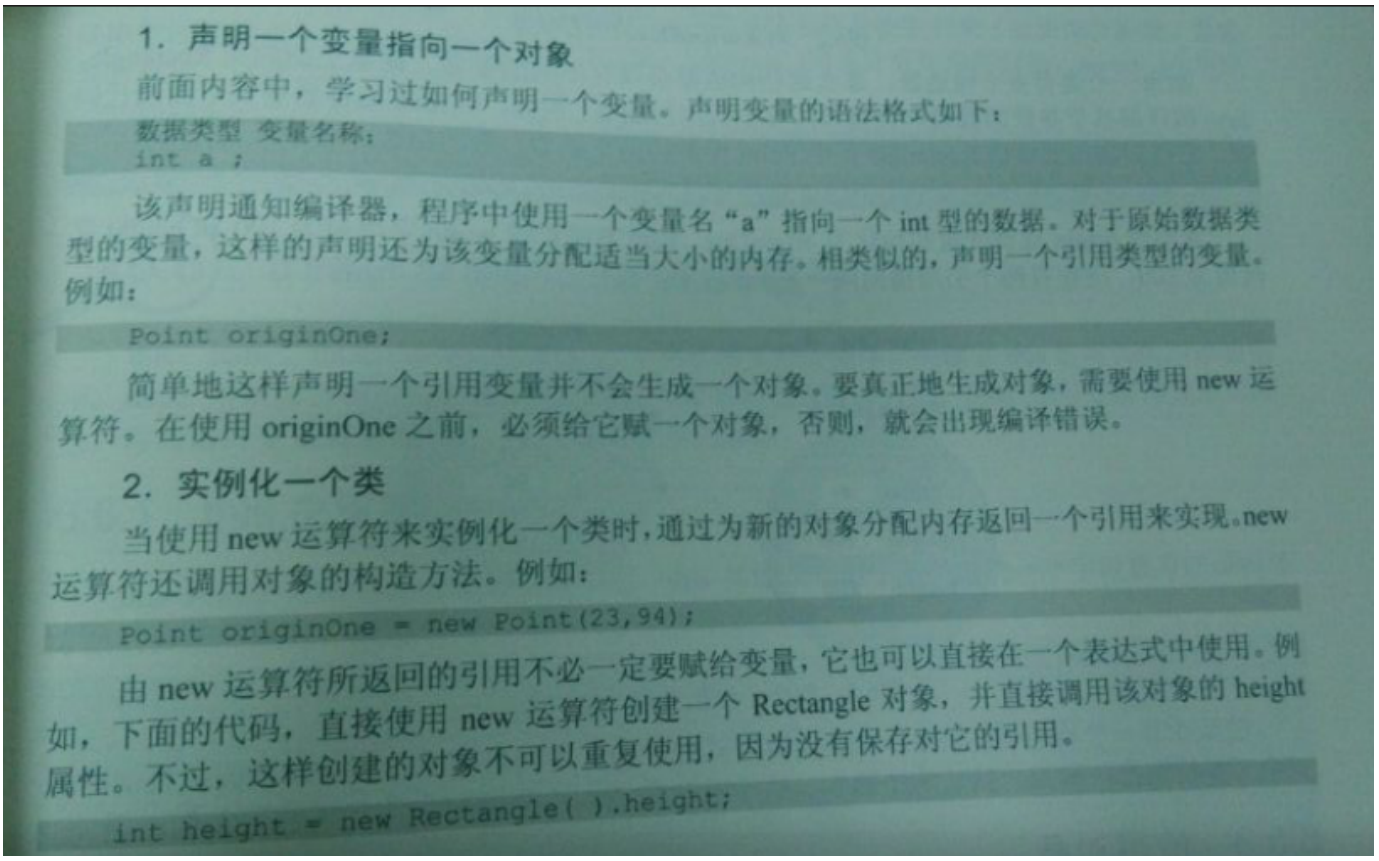
1.一个类只能继承一个父类。

2.在一个类中的成员变量被称为字段，在一个方法或代码块中的变量被成为局部变量。

3.**抽象类**：可能含有抽象方法，也可能没有，但是抽象类中可以含有非抽象的方法，抽象类不能用new运算符创建抽象类的实例对象，但是抽象类可以派
生子类。

4.Java接口：Java接口是一系列方法的声明，是一些方法特征的集合，一个接口只有方法的特征没有方法的实现，因此这些方法可以在不同的地方被不同的类实现，而这些实现可以具有不同的行为（功能）。

5.**抽象类与接口的对比**：接口可以被任何类实现，抽象类经常被子类化，并共享部分实现。在一个单独的抽象类中，提供大部分子类的共同点（即抽象类已经实现的部分），但是还有一些不同点，抽象方法只声明不实现，留给不同的子类根据自己的要求去实现。
抽象类可以包含有非static和final的字段，并且抽象类可以包含有实现的方法。这样的抽象类和接口很相似，但是抽象类提供部分方法的实现，其余的让子类来完成实现，如果一个抽象类只包含抽象方法的声明，那么应该将其声明为一个接口。



1．声明一个变量指向一个对象

前面内容中，学习过如何声明一个变量。声明变量的语法格式如下：

```
数据类型 变量名称;
int a ;
```

该声明通知编译器，程序中使用一个变量名“a”指向一个 int 型的数据。对于原始数据类型的变量，这样的声明还为该变量分配适当大小的内存。相类似的，声明一个引用类型的变量。例如：

```
Point originOne;
```

简单地这样声明一个引用变量并不会生成一个对象。要真正地生成对象，需要使用 new 运算符。在使用 originOne 之前，必须给它赋一个对象，否则，就会出现编译错误。

2．实例化一个类

当使用 new 运算符来实例化一个类时，通过为新的对象分配内存返回一个引用来实现。new 运算符还调用对象的构造方法。例如：

```
Point originOne = new Point(23,94);
```

由 new 运算符所返回的引用不必一定要赋给变量，它也可以直接在一个表达式中使用。例如，下面的代码，直接使用 new 运算符创建一个 Rectangle 对象，并直接调用该对象的 height 属性。不过，这样创建的对象不可以重复使用，因为没有保存对它的引用。

```
int height = new Rectangle( ).height;
```

6.**方法重写和方法隐藏后的修饰符：**

在子类中被重写的方法，其访问权限允许大于但是不允许小于被其重写的方法，例如：父类中一个protected的实例方法在子类中可以是public的

但是不可以是private的。

7.不允许将父类中的一个实例方法在子类中改变为类方法。意思就是，如果一个方法在父类中是static，则在子类中也应该是static的，如果一个方法在父类中是实例方法，则在子类中也必须是实例方法。