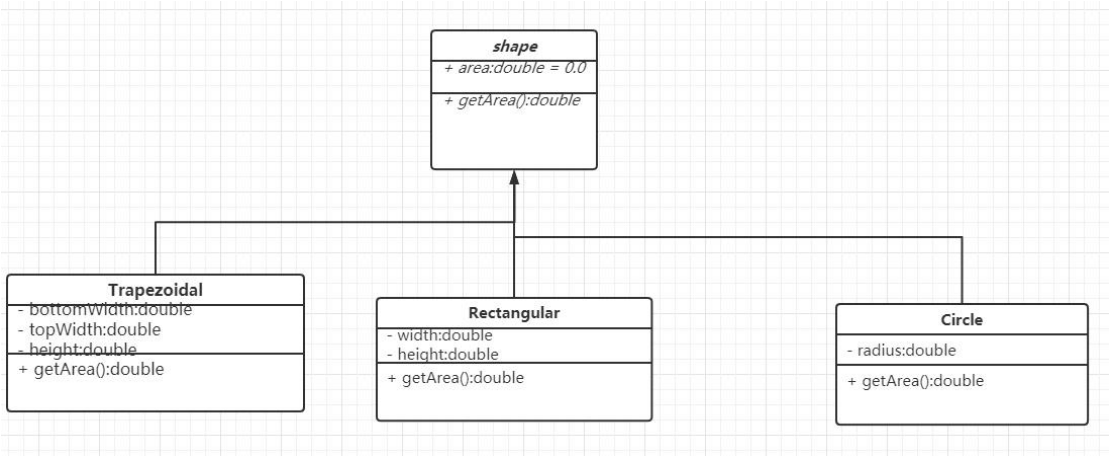
程序设计思路：
1.由于每种形状都有要获取面积的方法，因此可以对该方法进行抽象，放入一个 Shape 类中，该类中只有一个面积的成员属性和一个获取面积的抽象方法。
2.每种形状都继承该抽象类，并根据自己的面积计算方法实现该抽象类中的抽象方法 getArea();当有形状增加时只需要继承 Shape 方法并重写 getArea 方法即可。

3.创建一个 Controller 类，该类中有一个方法，接收一个字符串，分析该字符串并返回一个 Shape 类型。根据该字符串的第一个字符来判断是什么类型，并创建相应的对象返回，后续如果有增加的形状，在此方法中增加即可。

4.main 方法中逐行读入一个形状及其属性，每读入一行调用 Controller 中的方法获得一个具体的对象，并存放入一个保存 Shape 类型的数组 shapes 中（这里就使用了多态的形式，数组声明为 Shape 类型，但是保存的内容是其子类的具体实现类对象）。
5.计算钢板的利用率，遍历 shapes 数组，对每个对象调用 getArea 方法，累加面积值，最后所有累加值除于钢板的面积值。

总结：上面整个过程中需要修改 Controller 类中判断是什么类型的代码即可添加新的类型。



Shape 类型继承体系