**任务描述**

本关任务：通过使用Python的类继承等知识，完成一个简单的工厂模式。

**相关知识**

为了完成本关任务，你需要了解：

1. 工厂模式的思想
2. Python中类的继承
3. Python中类的静态方法

**设计模式**

设计模式是一套被反复使用的、多数人知晓的、经过分类编目的、代码设计经验的总结。使用设计模式是为了重用代码、让代码更容易被他人理解、保证代码可靠性。 毫无疑问，设计模式于己于他人于系统都是多赢的，设计模式使代码编制真正工程化，设计模式是软件工程的基石，如同大厦的一块块砖石一样。项目中合理地运用设计模式可以完美地解决很多问题，每种模式在现实中都有相应的原理来与之对应，每种模式都描述了一个在我们周围不断重复发生的问题，以及该问题的核心解决方案，这也是设计模式能被广泛应用的原因。

**简单工厂模式**

工厂模式是最常用的设计模式之一，它属于创建型模式，它提供了一种创建对象的最佳方式。在工厂模式中，我们在创建对象时不会对客户端暴露创建逻辑，并且是通过使用一个共同的接口来指向新创建的对象。

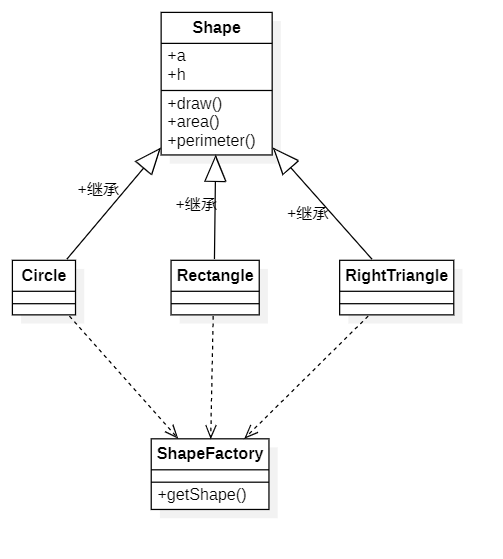
在简单工厂模式中，可以根据传递的参数不同，返回不同类的对象。简单工厂模式定义了一个类，这个类专门用于创建其他类的对象，这些被创建的类都有一个共同的父类。

**编程要求**

定义了Shape超类，Shape中有属性a和h表示图形的底和高，方法area()计算图形的面积，perimeter()计算图形的周长，draw()方法打印一个字符串，在超类中这些方法并未实现你需要在继承时在子类中实现这些方法。

Circle圆类、RightTriangle直角三角形类和Rectangle矩形类为Shape类的子类，你需要在子类中实现超类中的方法进行重写，根据子类自己的性质计算周长以及面积。

实现ShapeFactory工厂类，工厂类拥有一个静态方法getShape，该方法接受三个参数，classname参数为类名字符串，你需要根据不同的classname返回不同类的对象，如classname=Circle时，你需要返回一个Circle类对象，a,h为构造对象的参数。



**提示**

1. 超类Shape类你不需要改动，你只需实现Shape三个子类以及工厂类即可。
2. 实现Circle类时，默认属性a为圆的半径，h为0。
3. 子类需要自己的构造函数中调用父类的构造函数，并在构造函数中调用自己的draw方法，子类需要重写父类draw方法，draw方法中需要使用print打印字符串。比如Circle类应打印“Draw Circle!”其他子类类似。
4. 你无需重写Shape类的\_\_str\_\_方法。

**提示**

|  |
| --- |
| 测试函数：  def simple\_factory\_test():  c = ShapeFactory.getShape("Circle", 3, 0)  print(c)  b = ShapeFactory.getShape("Rectangle", 100, 600)  print(b)  s = ShapeFactory.getShape("RightTriangle", 3, 4)  print(s) |

预期输出：

|  |
| --- |
| Draw Circle!  a is 3.00,h is 0.00  Area is :28.27, Perimeter is :18.85  Draw Rectangle!  a is 100.00,h is 600.00  Area is :60000.00, Perimeter is :1400.00  Draw RightTriangle!  a is 3.00,h is 4.00  Area is :6.00, Perimeter is :12.00 |