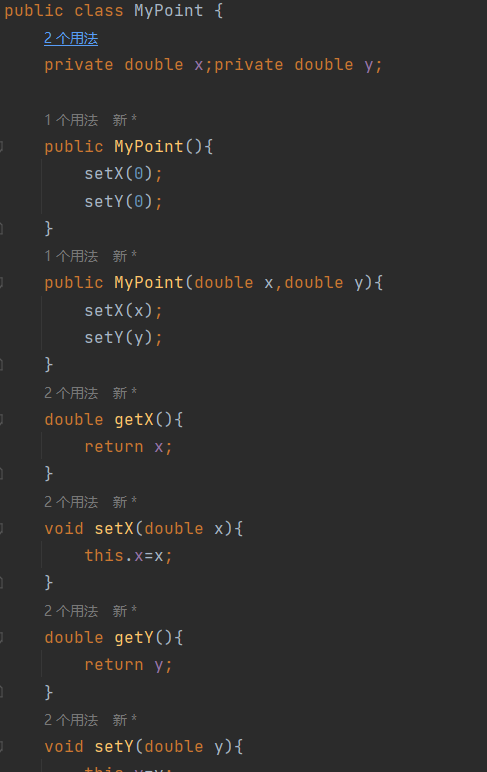
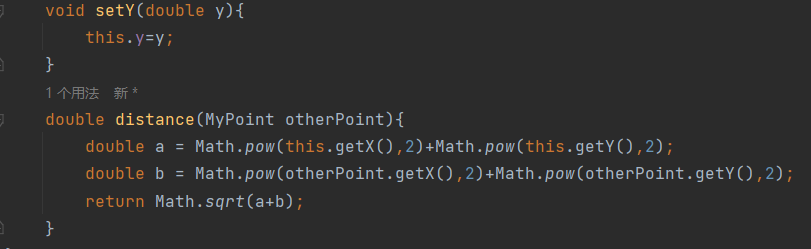
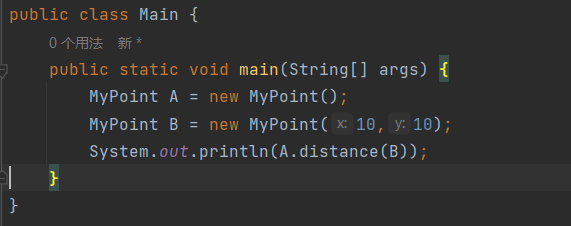
1.

MyPoint代码：

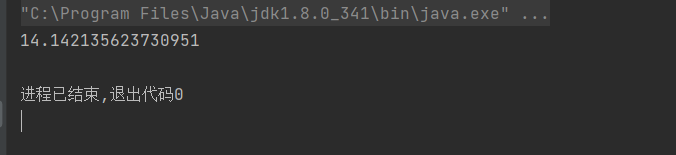




测试代码：

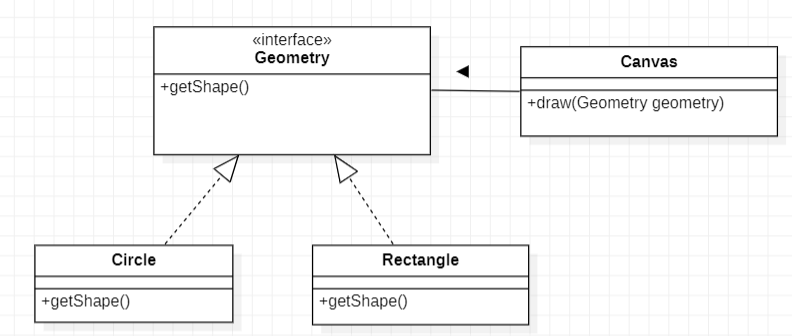


结果截图：



2.

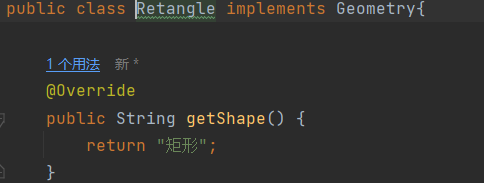
类图：

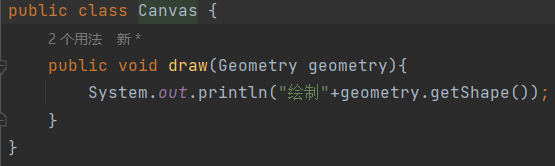


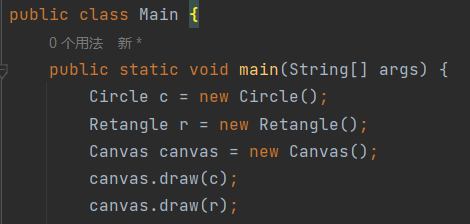
代码截图：

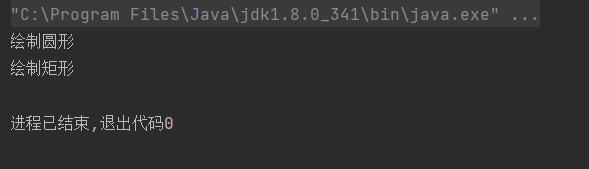
Geometry接口：

Circle类：

Retangle类：

Canvas类：

测试用例：

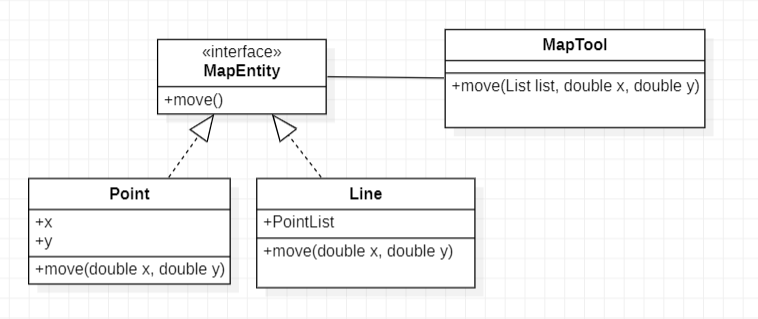
结果截图：

3.

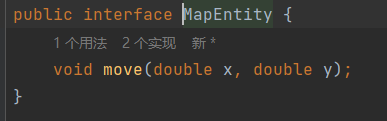
X应为将point和line抽象化的接口。

通过point和line对接口X的move的不同实现来降低MapTool中的move方法的耦合度。

类图：

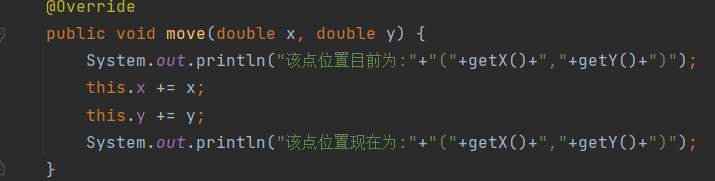


接口MapEntity：

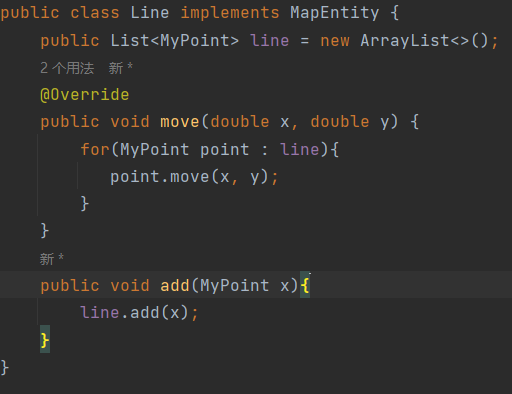


他有两个实现类，分别是point和line

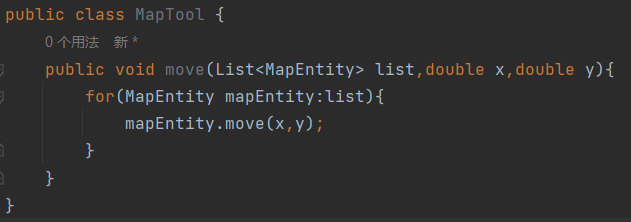
Point类：在第一题的基础上，令myPoint成为mapentity的实现类，然后加上重写函数move即可



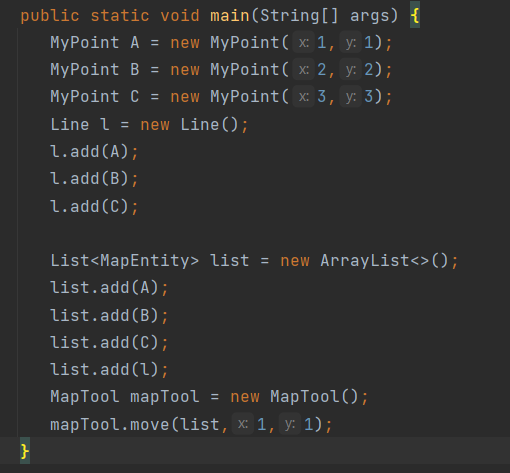
Line类：存储了一个有point组成的list，他的move函数是对list进行遍历，对每个point都调用对应的move方法。还包含了一个add方法，用来把point加入到line中



MapTool类：里面只有一个move方法，对参数list<mapEntity>进行move，原理和Line的move方法是一样的，只是其中的数据类型变成了mapEntity。



测试用例：



结果截图：

