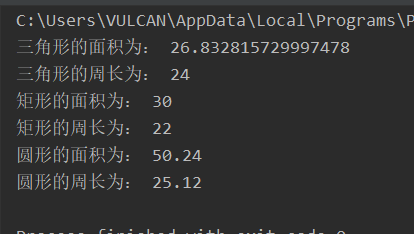
实验六 面向对象

1. 编程实现功能
2. 定义一个抽象类shape，在shape中定义求面积getArea()和周长getPerimeter（）的抽象方法。
3. 分别定义继承抽象类Shape的3个子类Triangle、Rectangle和Circle，在这3个子类中重写Shape中的方法getArea()和getPerimeter（）
4. 创建类Triangle、Rectangle和Circle的对象，对3个类中的方法进行调用测试。

import abc  
import math  
  
  
class shape(metaclass=abc.ABCMeta):  
 @abc.abstractmethod  
 def getArea(self):  
 pass  
  
 @abc.abstractmethod  
 def getPerimeter(self):  
 pass  
  
  
class Triangle(shape):  
 def \_\_init\_\_(self, a, b, c):  
 self.a = a  
 self.b = b  
 self.c = c  
  
 def getArea(self):  
 p = (self.a + self.b + self.c) / 2  
 S = math.sqrt(p \* (p - self.a) \* (p - self.b) \* (p - self.c))  
 print("三角形的面积为：", S)  
  
 def getPerimeter(self):  
 C = self.a + self.b + self.c  
 print("三角形的周长为：", C)  
  
  
class Rectangle(shape):  
 def \_\_init\_\_(self, a, b):  
 self.a = a  
 self.b = b  
  
 def getArea(self):  
 S = self.a \* self.b  
 print("矩形的面积为：", S)  
  
 def getPerimeter(self):  
 C = (self.a + self.b) \* 2  
 print("矩形的周长为：", C)  
  
  
class Circle(shape):  
 def \_\_init\_\_(self, r):  
 self.r = r  
  
 def getArea(self):  
 S = 3.14 \* self.r \* self.r  
 print("圆形的面积为：", S)  
  
 def getPerimeter(self):  
 C = 3.14 \* self.r \* 2  
 print("圆形的周长为：", C)  
  
  
tri = Triangle(7, 8, 9)  
tri.getArea()  
tri.getPerimeter()  
  
rec = Rectangle(5, 6)  
rec.getArea()  
rec.getPerimeter()  
  
cir = Circle(4)  
cir.getArea()  
cir.getPerimeter()



1. 设计一个“超市进销存管理系统”，要求如下：
2. 系统包括7种操作，分别是：1.查询所有商品：2.添加商品：3.修改商品：4.删除商品：5.卖出端口：6.汇总： -1.退出系统。
3. 选择操作序号“1”，显示所有商品。
4. 选择操作序号“2”，添加新的商品（包括商品名称、数量和进货价格）。
5. 选择操作序号“3“，修改商品。
6. 选择操作序号“4“，删除商品。
7. 选择操作序号“5“，卖出商品，包括商品名称、数量和售出价格。
8. 选择操作序号“6“，卖出商品，包括每种销售商品名称、数量、进货总价、销售总价等。
9. 选择操作序号“-1“，退出系统。

# 定义商品类  
class goods:  
 def \_\_init\_\_(self, name):  
 self.\_\_name = name  
  
 # 数量  
 @property  
 def count(self):  
 return self.\_\_count  
  
 @count.setter  
 def count(self, count):  
 self.\_\_count = count  
  
 # 进价  
 @property  
 def bid(self):  
 return self.\_\_bid  
  
 @bid.setter  
 def bid(self, bid):  
 self.\_\_bid = bid  
  
 # 售价  
 @property  
 def price(self):  
 return self.\_\_price  
  
 @price.setter  
 def price(self, price):  
 self.\_\_price = price  
  
 @property # 设置属性为只读.  
 def name(self):  
 return self.\_\_name  
  
  
list\_goods = []  
name = []  
good1 = goods("冰糖葫芦")  
good1.count = 5  
good1.bid = 3  
good1.price = 5  
  
  
list\_goods.append(good1)  
  
for i in list\_goods:  
 name.append(i.name)  
  
print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*超市系统\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  
print("1、显示所有商品")  
print("2、添加新的商品（包括商品名称、数量和进货价格）")  
print("3、修改商品")  
print("4、删除商品")  
print("5、卖出商品，包括商品名称、数量和售出价格")  
print("6、卖出商品，包括每种销售商品名称、数量、进货总价、销售总价等")  
print("-1、退出")  
  
while 1:  
 n = int(input("请输入你的选择："))  
 if n == 1:  
 for i in list\_goods:  
 print("商品名称：%s，数量：%d，售出单价：%.2f元" % (i.name, i.count, i.price))  
 elif n == 2:  
 name = str(input("请输入添加商品的名称："))  
 count = int(input("请输入添加商品的数量："))  
 bid = int(input("请输入添加商品的进价："))  
 price = int(input("请输入添加商品的售价："))  
 good = goods(name)  
 good.count = count  
 good.bid = bid  
 good.price = price  
 list\_goods.append(good)  
 for i in list\_goods:  
 print("商品名称：%s，数量：%d，售出单价：%.2f元" % (i.name, i.count, i.price))  
 print("商品添加成功！")  
 elif n == 3:  
 choice = str(input("请输入您要修改的商品名称："))  
 if choice in name:  
 for i in list\_goods:  
 count = int(input("请输入修改商品的数量："))  
 bid = int(input("请输入修改商品的进价："))  
 price = int(input("请输入修改商品的售价："))  
 i.count = count  
 i.bid = bid  
 i.price = price  
 print("修改成功！")  
 else:  
 print("超市中没有该商品")  
 elif n == 4:  
 choice = str(input("请输入您要删除的商品名称："))  
 if choice in name:  
 list\_goods.remove(i)  
 print("删除成功！")  
 else:  
 print("超市中没有该商品")  
  
 elif n == 5:  
 choice = str(input("请输入您要买的商品名称："))  
 if choice in name:  
 num = int(input("请输入您要买的数量："))  
 for i in list\_goods:  
 if i.name == choice:  
 if num <= i.count:  
 i.count -= num  
 totalPrice = num \* i.price  
 print("%s的数量还剩余：%d，售出价格为%.2f" % (i.name, i.count, totalPrice))  
 else:  
 print("商品剩余数量不足")  
 else:  
 print("超市中没有该商品")  
 elif n == 6:  
 choice = str(input("请输入您要买的商品名称："))  
 if choice in name:  
 num = int(input("请输入您要买的数量："))  
 for i in list\_goods:  
 if i.name == choice:  
 if num <= i.count:  
 total\_in = i.bid \* i.price  
 i.count -= num  
 totalPrice = num \* i.price  
 print("%s的数量还剩余：%d，售出总价为%.2f，购进总价为：%.2f"  
 % (i.name, i.count, totalPrice, total\_in))  
 else:  
 print("商品剩余数量不足")  
 else:  
 print("超市中没有该商品")  
 else:  
 exit()

