Objective:
- To understand abstract class

Task:
Implement the following classes and Create a Rectangle instance store in Shape reference invoke all abstract method

<>*Shape*
- name: String
+ Shape(name: String) /* accessor-mutator */ + abstract *area(): double* + abstract *perimeter(): double*

Rectangle extends Shape
- length: double - width: double
+ Rectangle(name: String, length: double, width: double) /* accessor-mutator */ + area(): double + perimeter(): double

Homework:
Implement the following classes and invoke discountedPrice() for object of each class.

<> Discountable
discountedPrice(price: double): double

PercentageDiscount
- percentage: double
+ PercentageDiscount(percentage: double) + getPercentage(): double + setPercentage(percentage: double): void + discountedPrice(price: double): double

ThresholdDiscount
- threshold: double - discount: double
+ ThresholdDiscount(threshold: double, discount: double) + getThreshold(): double + getDiscount(): double + setThreshold(threshold: double): void + setDiscount(discount: double): void + discountedPrice(price: double): double