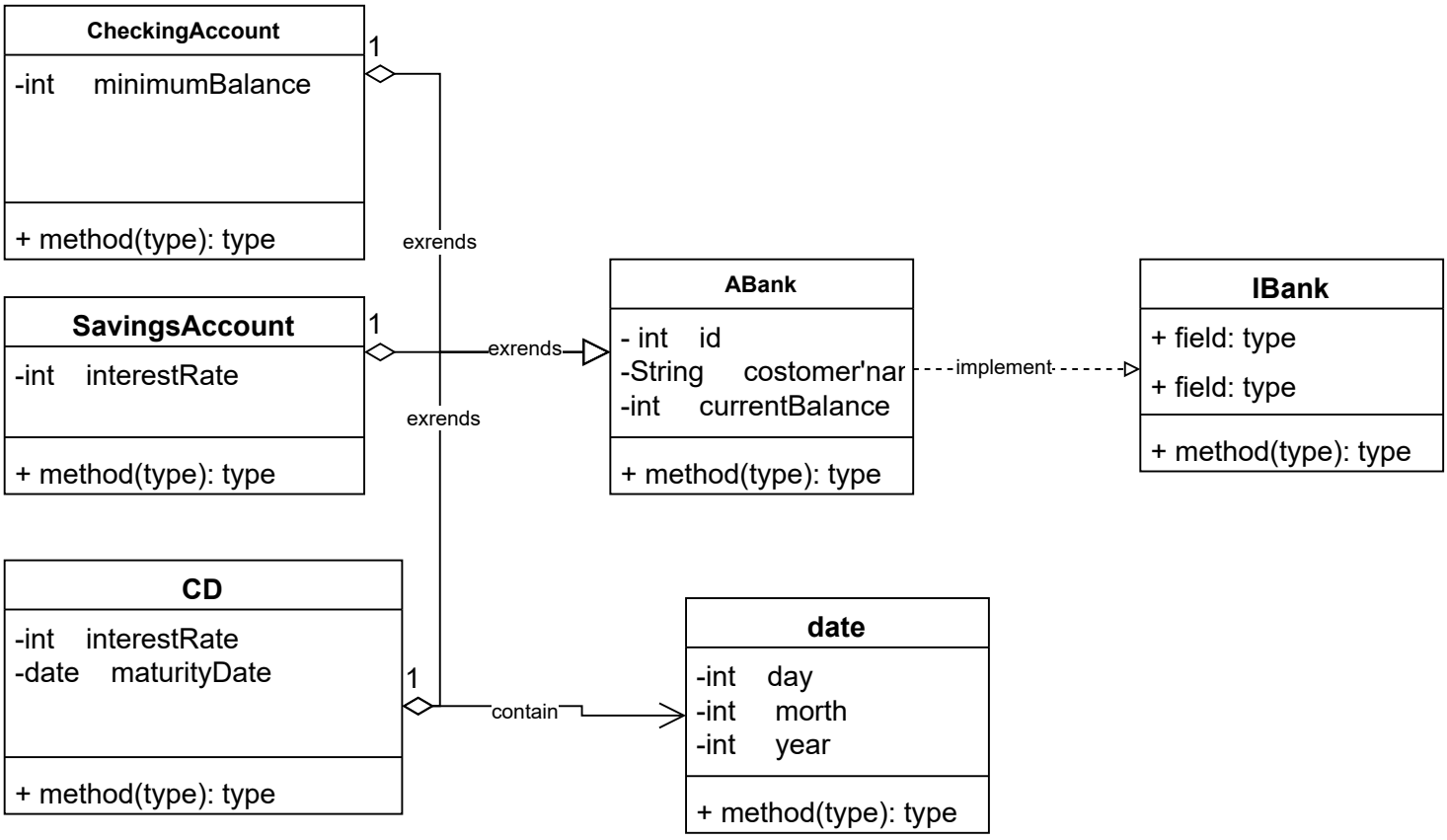
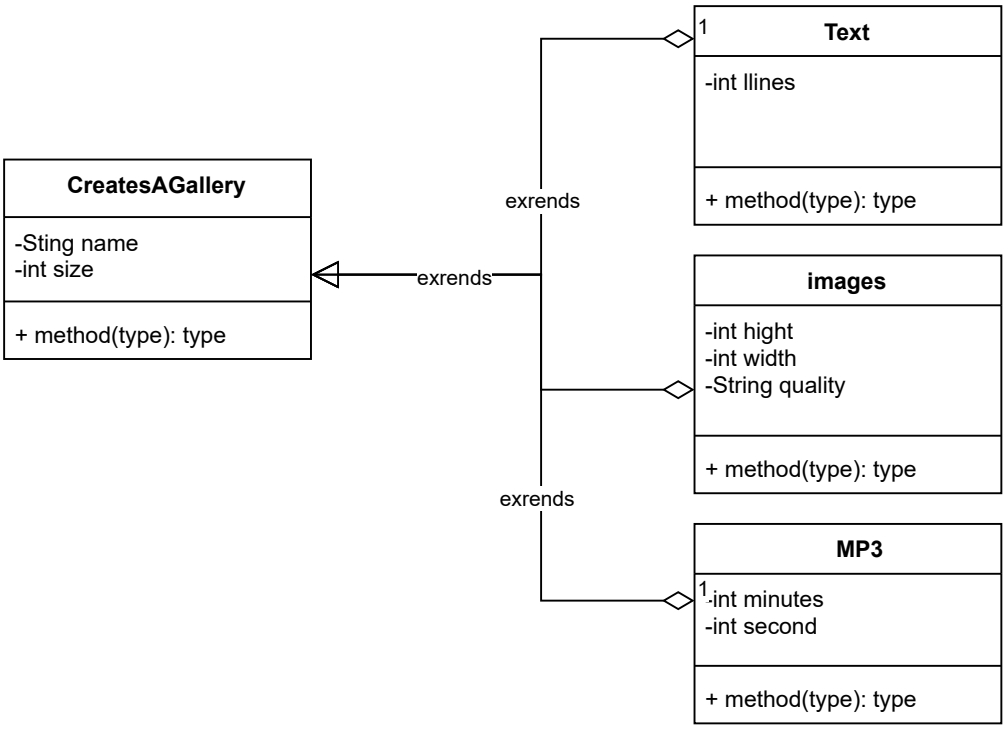
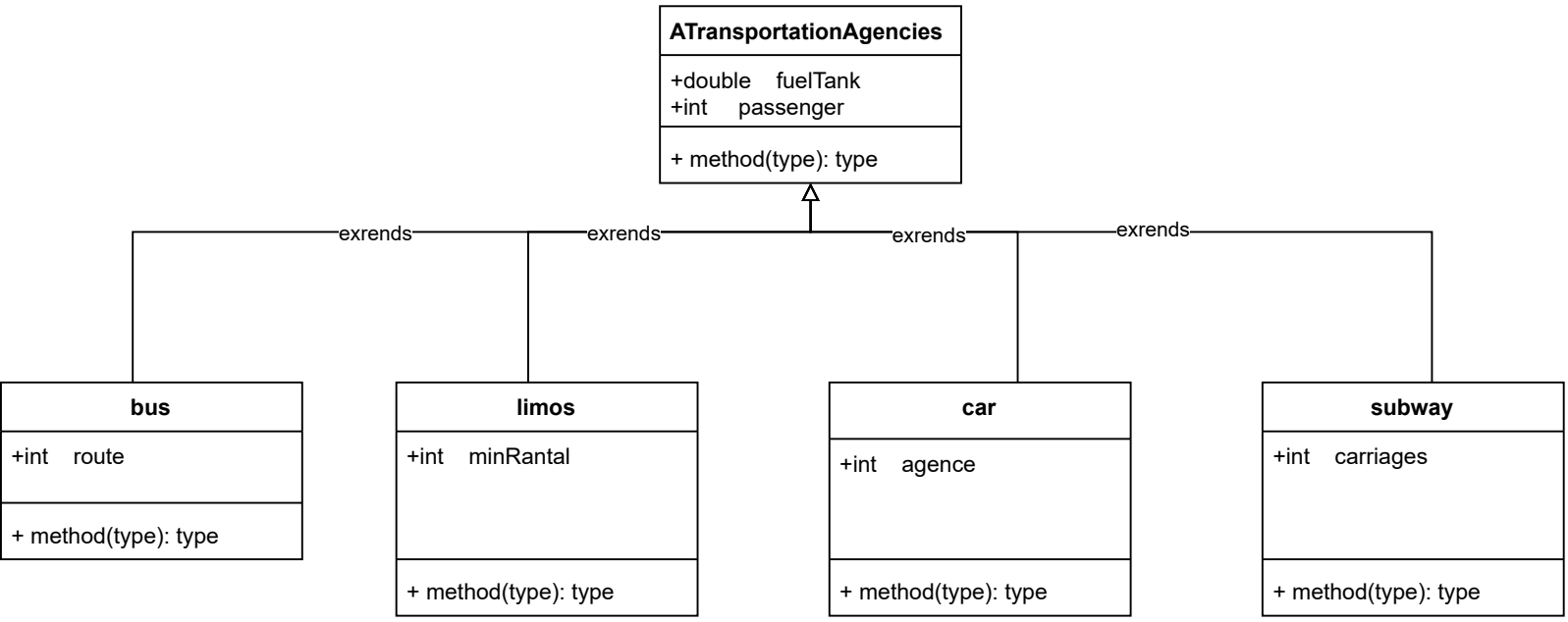
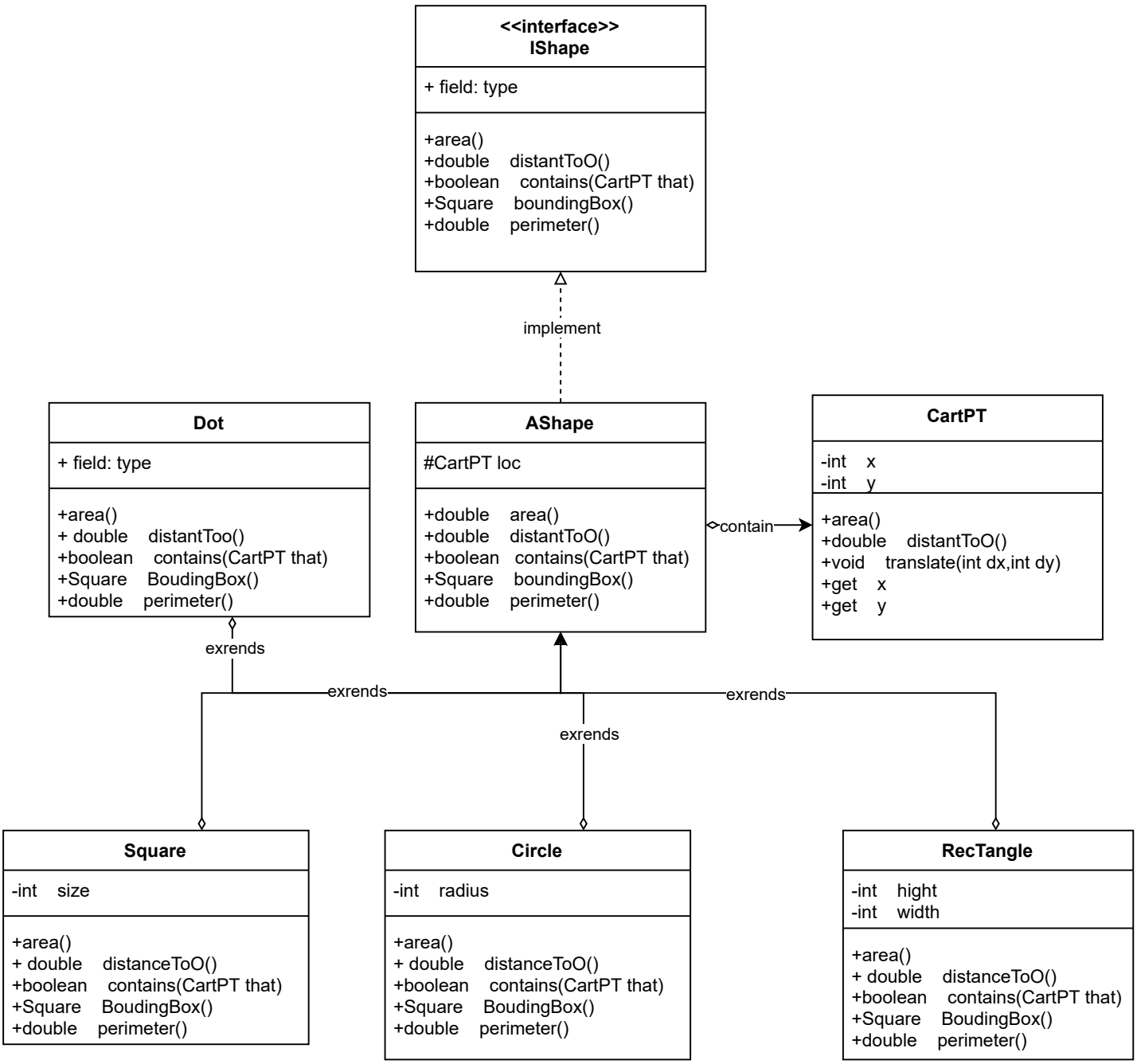

CheckingAccount
-int minimumBalance
+ method(type): type
1
exrends
SavingsAccount
-int interestRate
+ method(type): type
1
exrends
exrends
ABank
- int id
-String costomer'nar
-int currentBalance
+ method(type): type
implement
IBank
+ field: type
+ field: type
+ method(type): type
CD
-int interestRate
-date maturityDate
+ method(type): type
1
contain
date
-int day
-int morth
-int year
+ method(type): type


CreatesAGallery
-Sting name
-int size
+ method(type): type
exrends
exrends
exrends
1
Text
-int llines
+ method(type): type
images
-int hight
-int width
-String quality
+ method(type): type
MP3
1
-int minutes
-int second
+ method(type): type


ATransportationAgencies
+double fuelTank
+int passenger
+ method(type): type
exrends
exrends
exrends
exrends
bus
+int route
+ method(type): type
limos
+int minRantal
+ method(type): type
car
+int agence
+ method(type): type
subway
+int carriages
+ method(type): type



4.4.1 Design an extension for the class hierarchy of shapes that deals with Rectangles.
The extension should cope with all the abstract methods in AShape

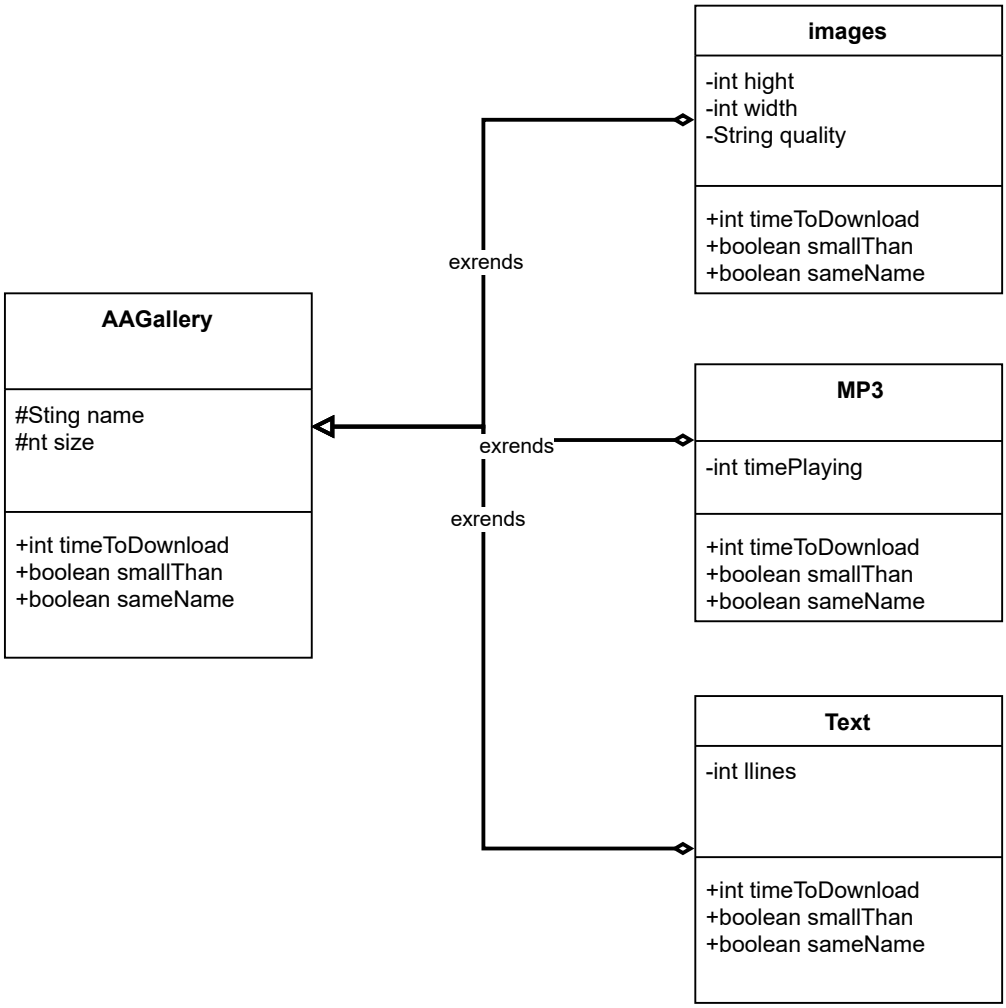
4.4.2 Design an extension for the class hierarchy of shapes so that a program can request the length of the perimeter for each shape in the hierarchy

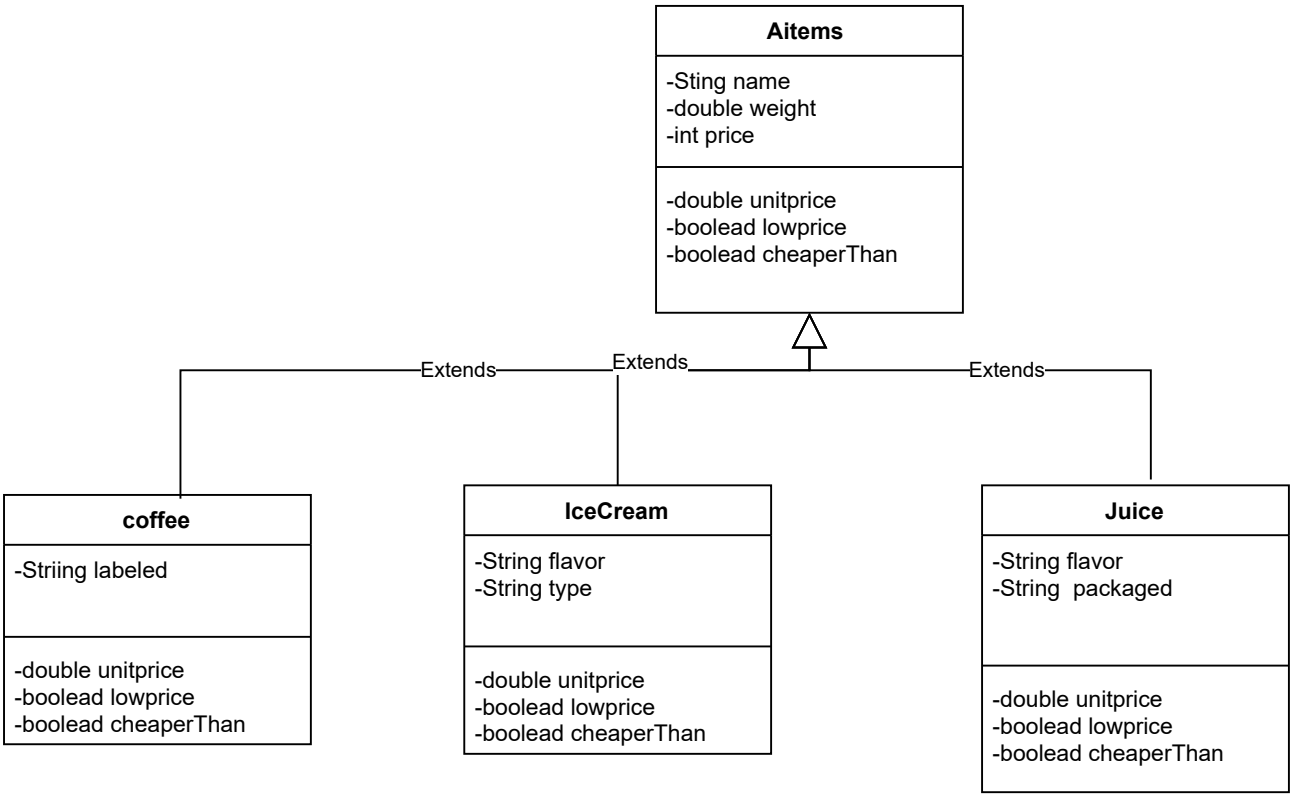
4.4.3 Extended
Compute the bounding box the class hierarchy of shapes that deals with Rectangles. The extension should cope with all the abstract methods in AShape

Develop the following methods for this program:
timeToDownload, which computes how long it takes to download a file at some network connection speed, typically given in bytes per second;

smallerThan, which determines whether the file is smaller than some given maximum size that can be mailed as an attachment;

sameName, which determines whether the name of a file is the same as some given name.

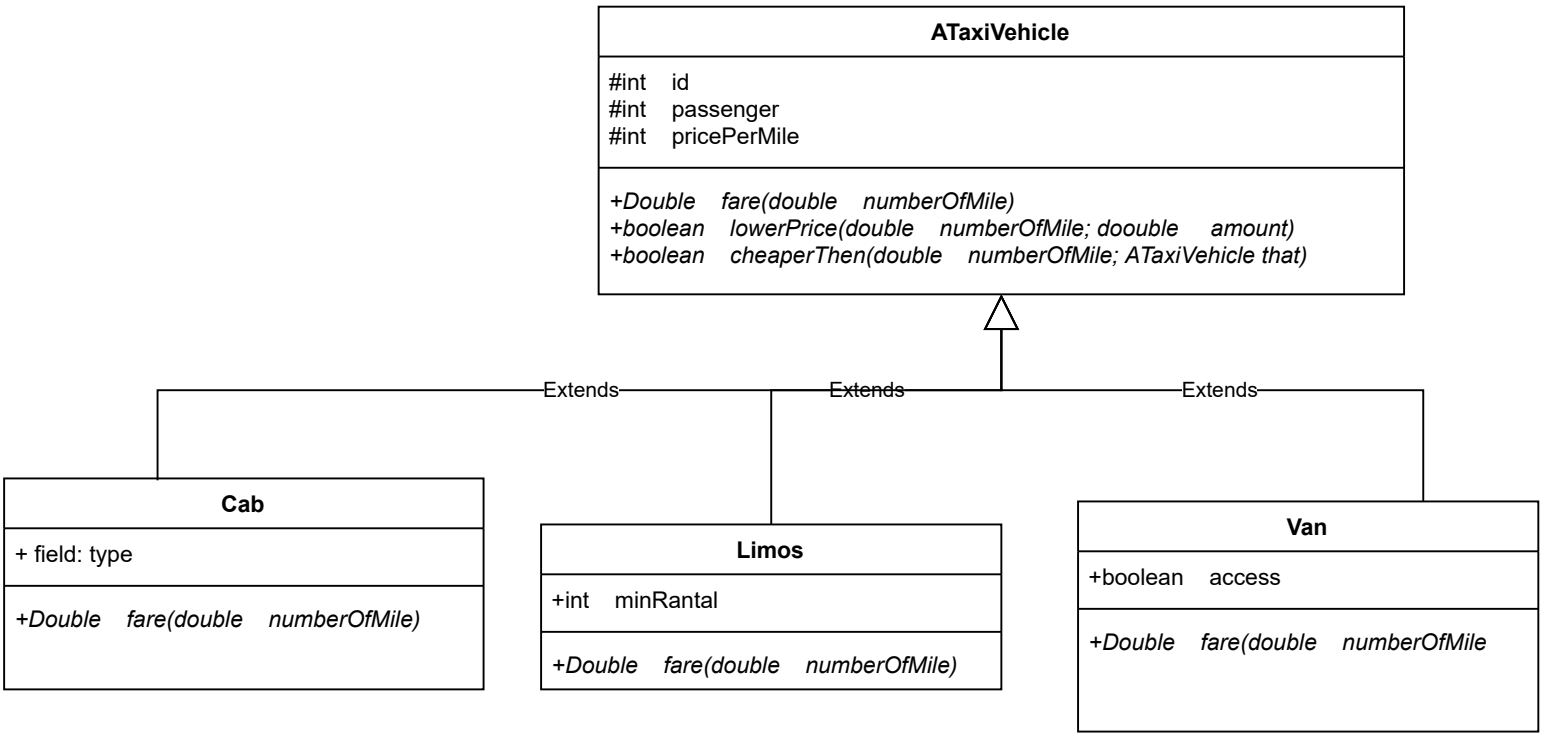




unitPrice, which computes the unit price (cents per gram) of some grocery item;

lowerPrice, which determines whether the unit price of some grocery item is lower than some given amount;

cheaperThan, which determines whether a grocery item is cheaper than some other, given one in terms of the unit



Add the following methods to the appropriate classes in the hierarchy:

fare, which computes the fare for a vehicle, based on the number of miles traveled, and using the following formulas for different vehicles:

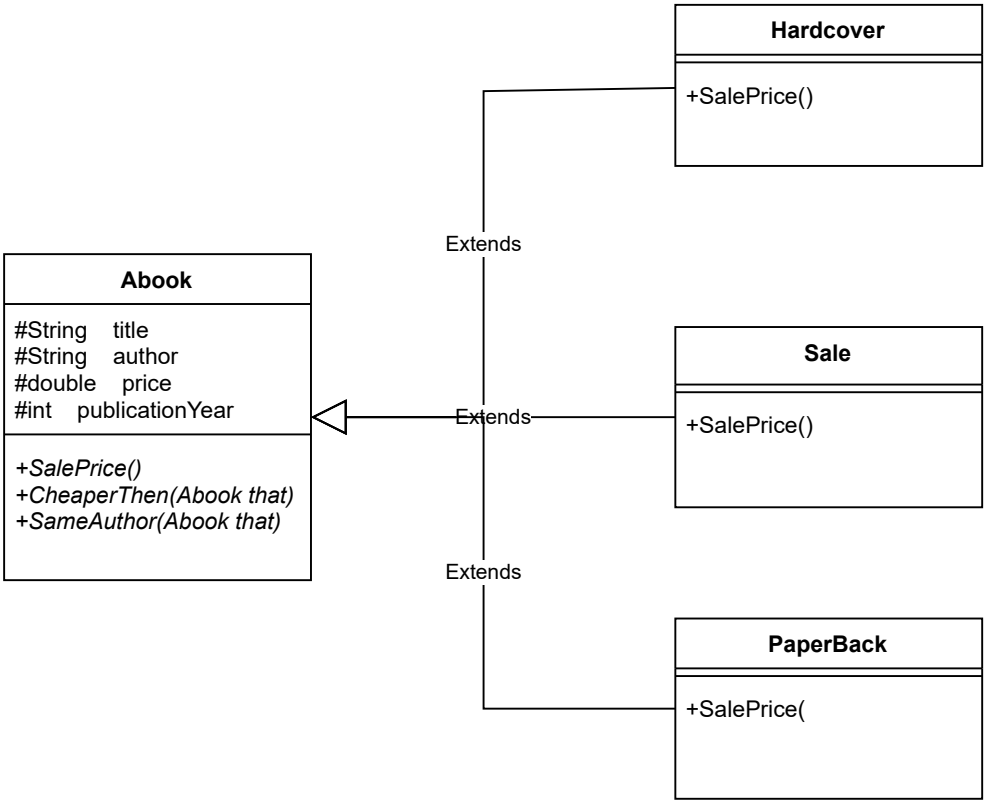
passengers in a cab just pay flat fee per mile

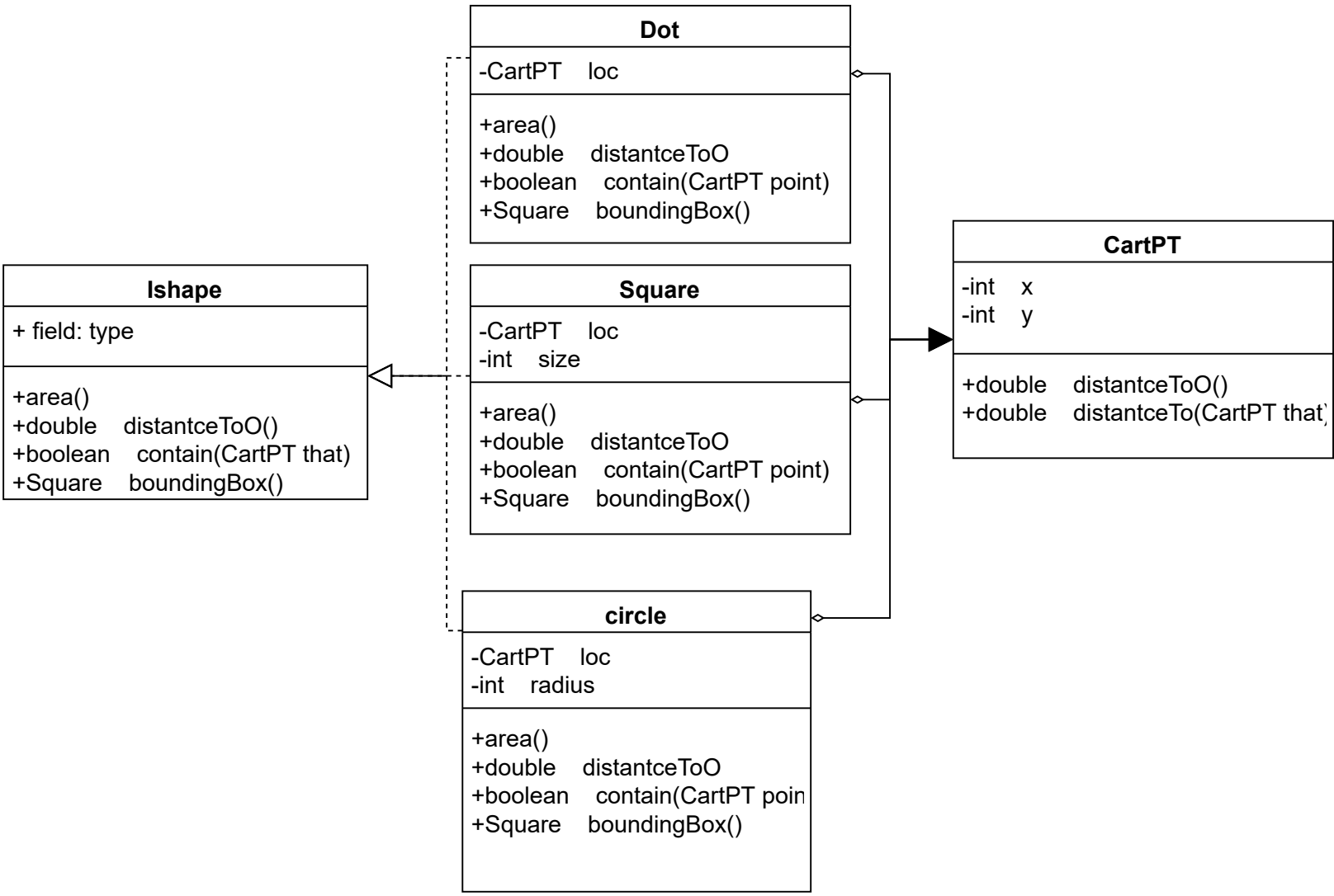
passengers in a limo must pay at least the minimum rental fee, otherwise they pay by the mile

passengers in a van pay $1.00 extra for each passenger

lowerPrice, which determines whether the fare for a given number of miles is lower than some amount;

cheaperThan, which determines whether the fare in one vehicle is lower than the fare in another vehicle for the same number of miles.


Hardcover
+SalePrice()
Extends
Abook
#String title
#String author
#double price
#int publicationYear
+SalePrice()
+CheaperThen(Abook that)
+SameAuthor(Abook that)
Sale
Extends
+SalePrice()
Extends
PaperBack
+SalePrice(



Compute the area of a shape

Compute the distance of a shape to the origin

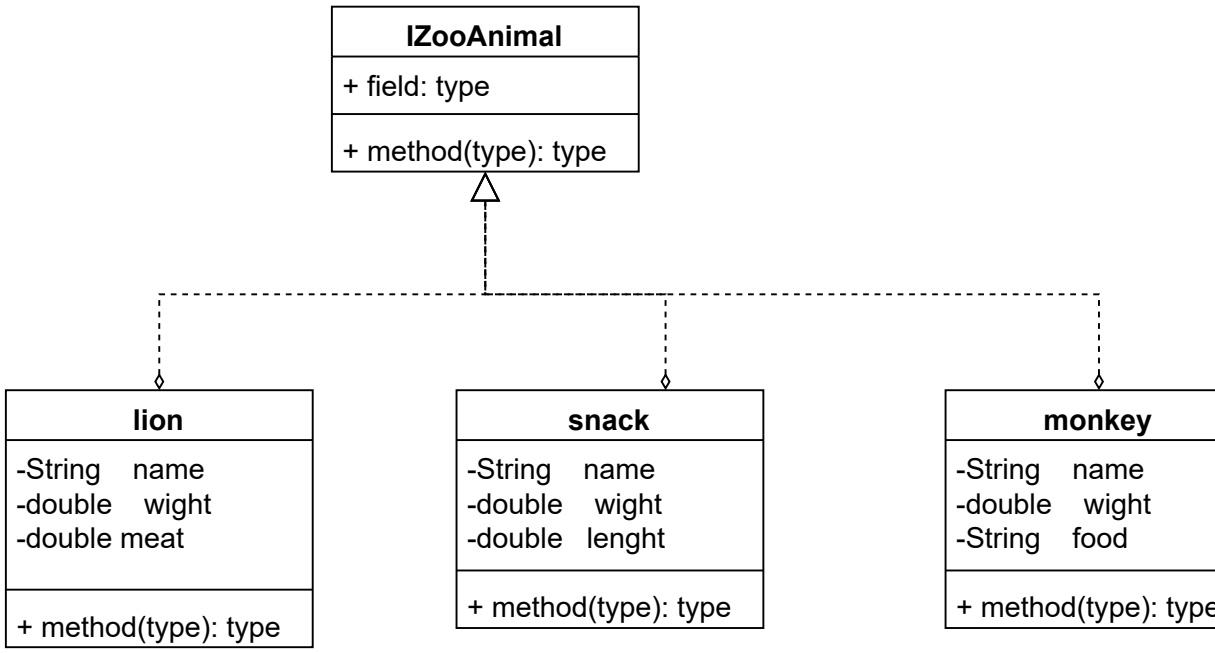
Determine whether some point is inside the shape

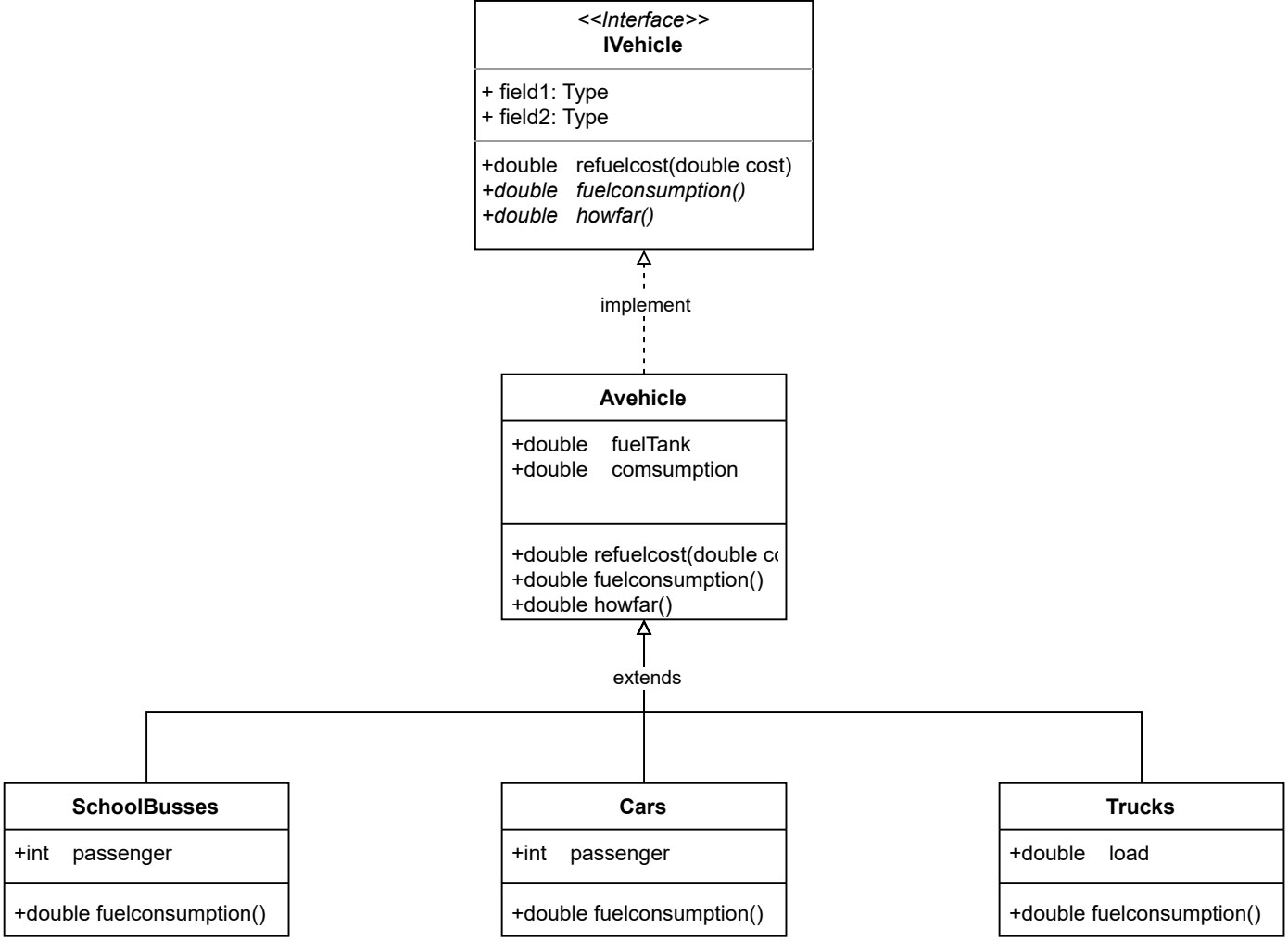
Compute the bounding box of a shape

All of these methods clearly work with shapes in general but may have to compute different results depending on the concrete shape on which they are invoked

For example, a Dot has no true area; a Square's area is computed differently from a Circle's area

In an object-oriented language, we can express this requirement with the addition of a method signature to the IShape interface


IZooAnimal
+ field: type
+ method(type): type
lion
-String name
-double wight
-double meat
+ method(type): type
snack
-String name
-double wight
-double lenght
+ method(type): type
monkey
-String name
-double wight
-String food
+ method(type): type


<<Interface>>
IVehicle
+ field1: Type
+ field2: Type
+double refuelcost(double cost)
+double fuelconsumption()
+double howfar()
implement
Avehicle
+double fuelTank
+double comsumption
+double refuelcost(double c
+double fuelconsumption()
+double howfar()
extends
SchoolBusses
+int passenger
+double fuelconsumption()
Cars
+int passenger
+double fuelconsumption()
Trucks
+double load
+double fuelconsumption()