Exercises 5.1 – 5.4
Containment in Union

Exercises 5.1

- Develop a program that assists real estate agents. The program deals with listings of available houses. The information about a house includes its kind, the number of rooms, the asking price, and its address. An address consists of a house number, a street name, and a city.
  - Ranch, 7 rooms, $375,000, 23 Maple Street, Brookline
  - Colonial, 9 rooms, $450,000, 5 Joye Road, Newton
  - Cape, 6 rooms, $235,000, 83 Winslow Road, Waltham
- Make examples of listings.
  Develop a data definition for listings of houses.
  - Implement the definitionwith classes.
    Translate the examples into objects

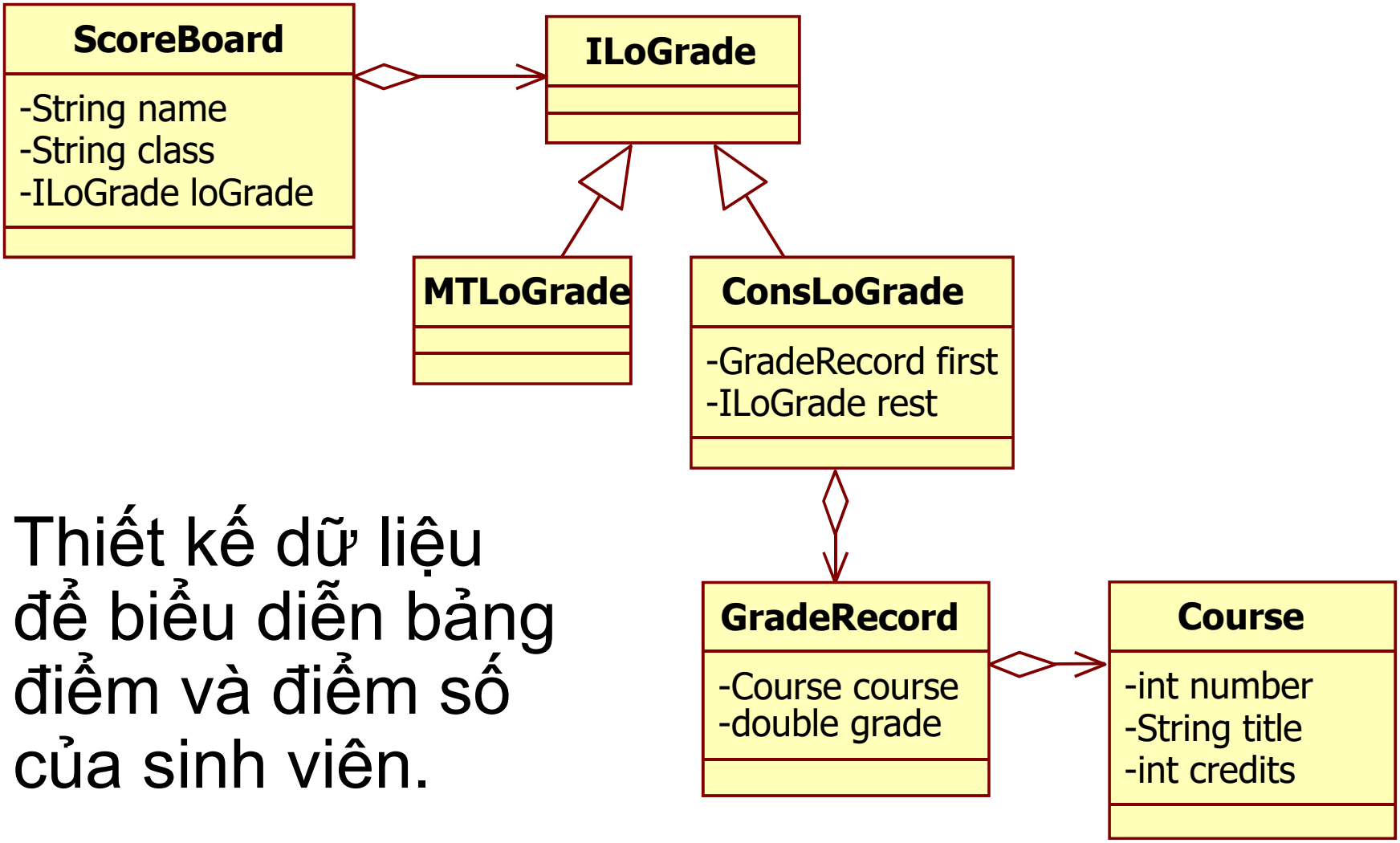
Exercises 5.2

- Design a program that assists a bookstore manager with reading lists for local schools. . . For each book, the program should track the book's title, its price, its year of publication, and the author's name, …
  - Daniel Defoe, Robinson Crusoe, $15.50, 1719;
  - Joseph Conrad, Heart of Darkness, $12.80, 1902;
  - Pat Conroy, Beach Music, $9.50, 1996.
- Develop a class diagramfor a list of books (by hand). Translate the diagram into classes.
- Create two lists of books that contain one or more of your favorite books

Excercise 5.3

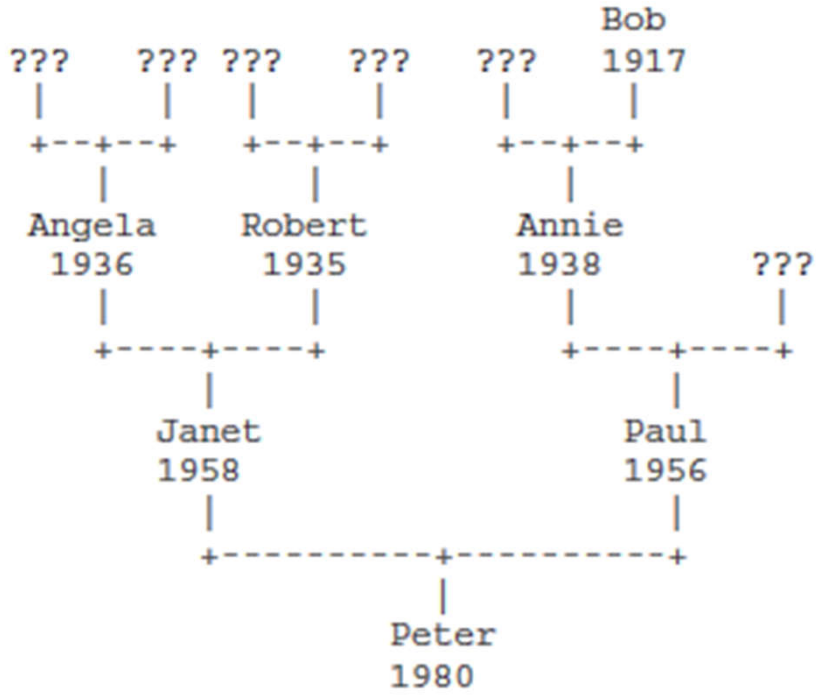
- Thông tin về điểm số của mỗi sinh viên được cho trong một bảng điểm. Mỗi bảng điểm (**ScoreBoard**) bao gồm tên sinh viên (**name**), khóa học (**class**), và một danh sách điểm số các môn học của sinh viên. Thông tin về điểm số (**GradeRecord**) của sinh viên bao gồm mã số môn học (**number**), tên môn học (**title**), số tín chỉ (**credits**) và điểm số (**grade**).
  - Ví dụ: một bảng điểm của sinh viên **Tran Van Hoa**, khóa **2009** gồm các mục điểm số:
    - 211, "Database Fundamentals", 3, 7.5
    - 220, "Basic Programming", 2, 5.0
    - 690, "Algorithms", 4, 7.0
    - 721, "Data Structure", 4, 8.0

Excercise 5.3 (cont)



Thiết kế dữ liệu để biểu diễn bảng điểm và điểm số của sinh viên.

Exercises 5.4



Develop a programthat helps with recording a person's ancestor tree. Specifically, for each person we wish to remember the person's nam and year of birth, in addition to the ancestry on the father's and the mother's side, if it is available.

The tree on the left is an example; the nodes with "???" indicate where the genealogist couldn't find any information.

- Develop the class diagram and the class definitions to represent ancestor family trees.
- Also draw your family's ancestor tree as far as known and represent it as an object.

Exercises 5.5 – 5.8 Containment in Union and Methods

Exercise 5.5

5.5.1 Suppose the requirements for the program that tracks a runner's log includes this request: ... The runner wants to know the total distance run in a given month...

- Design the method that computes this number

5.5.2 Suppose the requirements for the program that tracks a runner's log includes this request: ... A runner wishes to know the maximum distance ever run ...

- Design the method that computes this number. Assume that the method produces 0 if the log is empty.

Exercise 5.5 (cont)

5.5.3 Suppose the requirements for the program that tracks a runner's log includes this request:

... The runner would like to see the log with entries ordered according to the pace computed in minutes per mile in each run, from the fastest to the slowest ...

- Design this sorting method.
  Hint: Don't forget to design methods for auxiliary tasks.

Exercise 5.6

- A phone directory combines names with phone numbers. Develop a data definition for phone records and directories.
  Develop the methods:
  - `whoseNumber`, which determines the name that goes with some given phone number and phone directory.
  - `phoneNumber`, which determines the phone number that goes with some given name and phone directory

Exercise 5.7

Design a data representation for shopping lists.

Start from the class of grocery items developed in exercise 4.6. Add the following methods:

- howMany, which computes the number of items on the shopping list;
- brandList, which produces the list of all brand names; and
- highestPrice, which determines the highest unit price among all items in the shopping list.

Exercise 5.8

Add methods to `Inventory` use case

5.5.1 Define the method `averagePrice`. It computes the average price of toys in `Inventory`. The average is the total of all prices divided by the number of toys

5.5.2 Develop the method `replaceName`, which consumes a list of toy and replaces all occurrences of “robot” with “r2d2” and otherwise retains the toy descriptions in the same order.

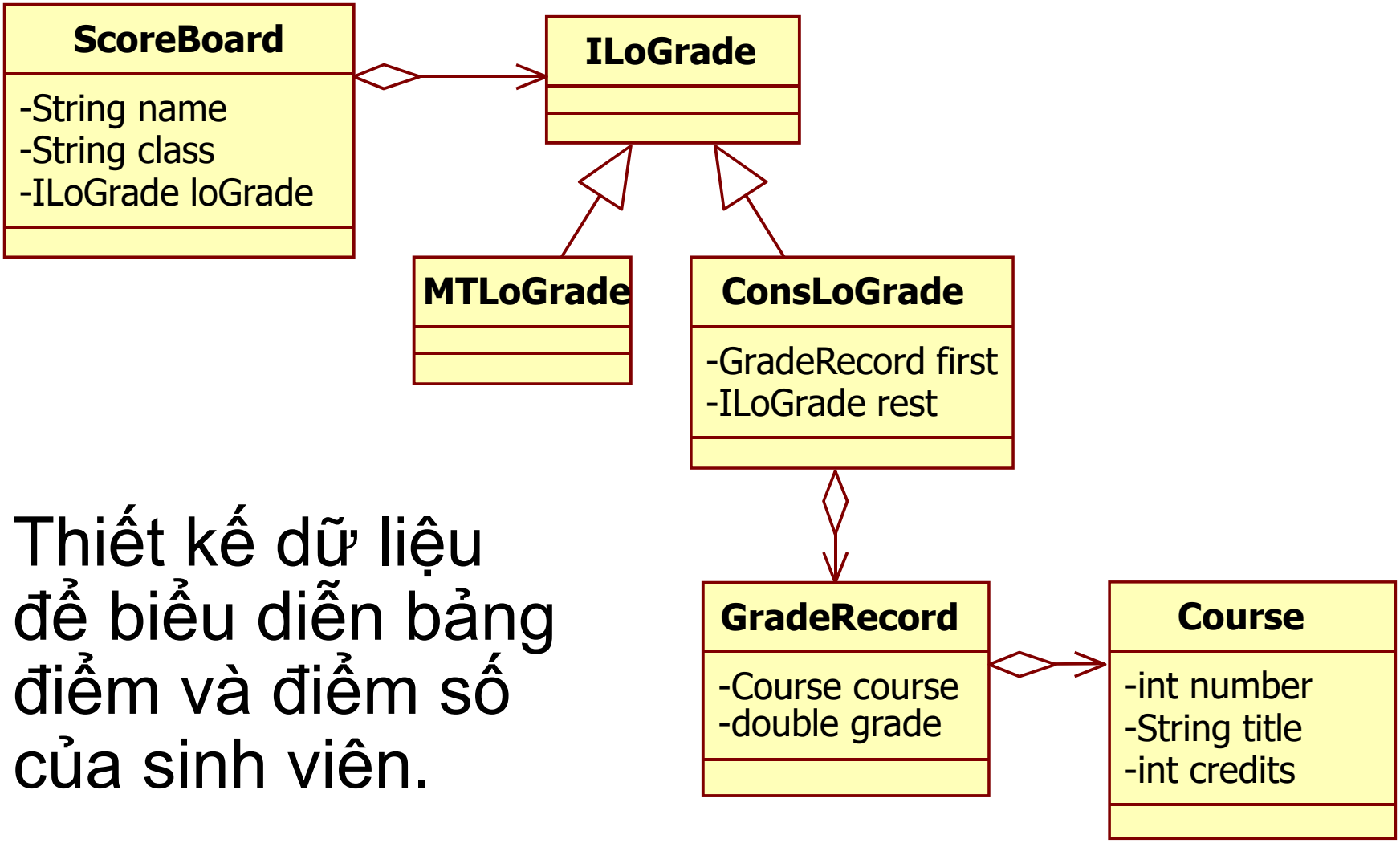
5.5.3 Develop the method `eliminate`. The method consumes a string, called `toyOfName` and produces a list of toys that contains all components of list with the exception of the toy whose name matches `toyOfName`.

Excercise 5.9

Lấy lại bài tập 5.3 biểu diễn bảng điểm và điểm số của sinh viên trong và thiết kế các phương thức sau:

- Viết phương thức **howManyCredits** để tính tổng số tín chỉ trong bảng điểm mà sinh viên đã đạt được.
- Viết phương thức **gradeAverage** để tính điểm trung bình của sinh viên bằng tổng của tích điểm số từng môn với số tín chỉ chia cho tổng số tín chỉ.
- Viết phương thức **sortByGradeDec** để sắp xếp bảng điểm số của sinh viên theo thứ tự điểm giảm dần.
- Viết phương thức **greaterThanList** để trả về danh sách mục điểm số của sinh viên có điểm lớn hơn một giá trị cho trước.

ScoreBoard class diagram



Thiết kế dữ liệu để biểu diễn bảng điểm và điểm số của sinh viên.

Relax &

…Do Exercises …

5.5.1 `miles()` for `ILog`

```
public interface ILog {
    ...

    // to compute the total number of miles
    // recorded in this log for the given month and year
    public double miles(int month, int year);
}
```

- Q: Develop some examples to test the `miles()` method

Examples to test `miles()`

```
Entry e1 = new Entry(new Date(5, 5, 2005), 5.0, 25, "Good");
Entry e2 = new Entry(new Date(6, 6, 2005), 3.0, 24, "Tired");
Entry e3 = new Entry(new Date(23, 6, 2005), 26.0, 156, "Great");

ILog l0 = new MTLog();
ILog l1 = new ConsLog(e1, l0);
ILog l2 = new ConsLog(e2, l1);
ILog l3 = new ConsLog(e3, l2);

l0.miles(6, 2005) → should be 0.0
l1.miles(6, 2005) → should be 0.0
l2.miles(6, 2005) → should be 3.0
l3.miles(6, 2005) → should be 29.0
```

Q: Implement `miles()` in `MTLog` and `ConsLog`

`miles()` for `MTLog`

```
public class MTLog implements ILog {
    // ...

    public double miles(int month, int year) {
        return 0.0;
    }
}
```

`miles()` for `ConsLog`

```
public class ConsLog implements ILog {
   private Entry first;
   private ILog  rest;
   // ...

   public double miles(int month, int year) {
      if (this.first.sameMonthInAYear(month, year))
         return this.first.getDistance() +
                this.rest.miles(month, year));
      else
         return this.rest.miles(month, year);
   }
}
```

5.5.2 `maxDistance()` for `ILog`

```
public interface ILog {
    ...

    // to compute the total number of miles
    // recorded in this log for the given month and year
    public double miles(int month, int year);

    // to compute the maximize distance
    // recorded in this log
    public double maxDistance();
}
```

- Q: Develop some examples to test the `maxDistance()` method

Examples to test `maxDistance()`

```
Entry e1 = new Entry(new Date(5, 5, 2005), 5.0, 25, "Good");
Entry e2 = new Entry(new Date(6, 6, 2005), 3.0, 24, "Tired");
Entry e3 = new Entry(new Date(23, 6, 2005), 26.0, 156, "Great");

ILog l0 = new MTLog();
ILog l1 = new ConsLog(e1, l0);
ILog l2 = new ConsLog(e2, l1);
ILog l3 = new ConsLog(e3, l2);

l0.maxDistance() → should be 0.0
l1.maxDistance() → should be 5.0
l2.maxDistance() → should be 5.0
l3.maxDistance() → should be 26.0
```

Q: Implement `maxDistance()` in `MTLog` and `ConsLog`

maxDistance() for MTLog

```
public class MTLog implements ILog {
    // ...

    public double maxDistance() {
        return 0.0;
    }
}
```

maxDistance() for ConsLog

```
public class ConsLog implements ILog {
   private Entry first;
   private ILog  rest;
   // ...

   public double maxDistance() {
      return Math.max(this.first.getDistance(),
                      this.rest.maxDistance());
   }
}
```

Other implement ation

`maxDistance()`

```
public class MTLog implements ILog {
    // ...

    public double maxDistance() {
        return 0.0;
    }

    public double maxDistance() {
        return 0.0;
    }
}
```

maxDistance() for ConsLog

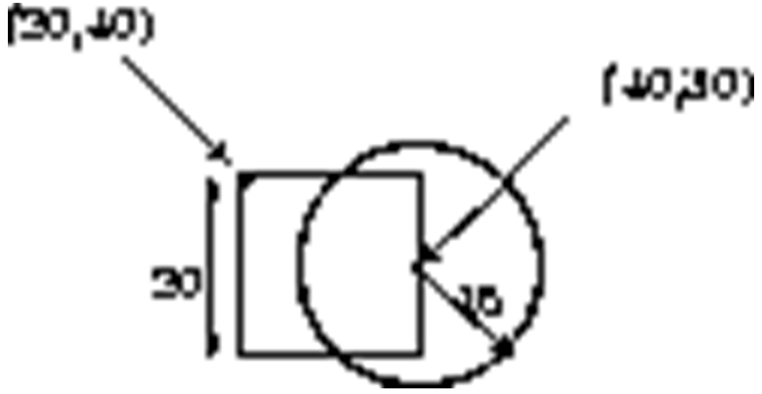
```
public class ConsLog implements ILog {
   private Entry first;
   private ILog  rest;
   // ...

   public double maxDistance() {
      return maxHelper(this.first.getDistance());
   }

   public double maxHelper(double maxCurrent) {
      if (this.first.getDistance() > maxCurrent)
         return this.rest.maxHelper(this.first.getDistance());
      else
         return this.rest.maxHelper(maxCurrent);
   }
}
```

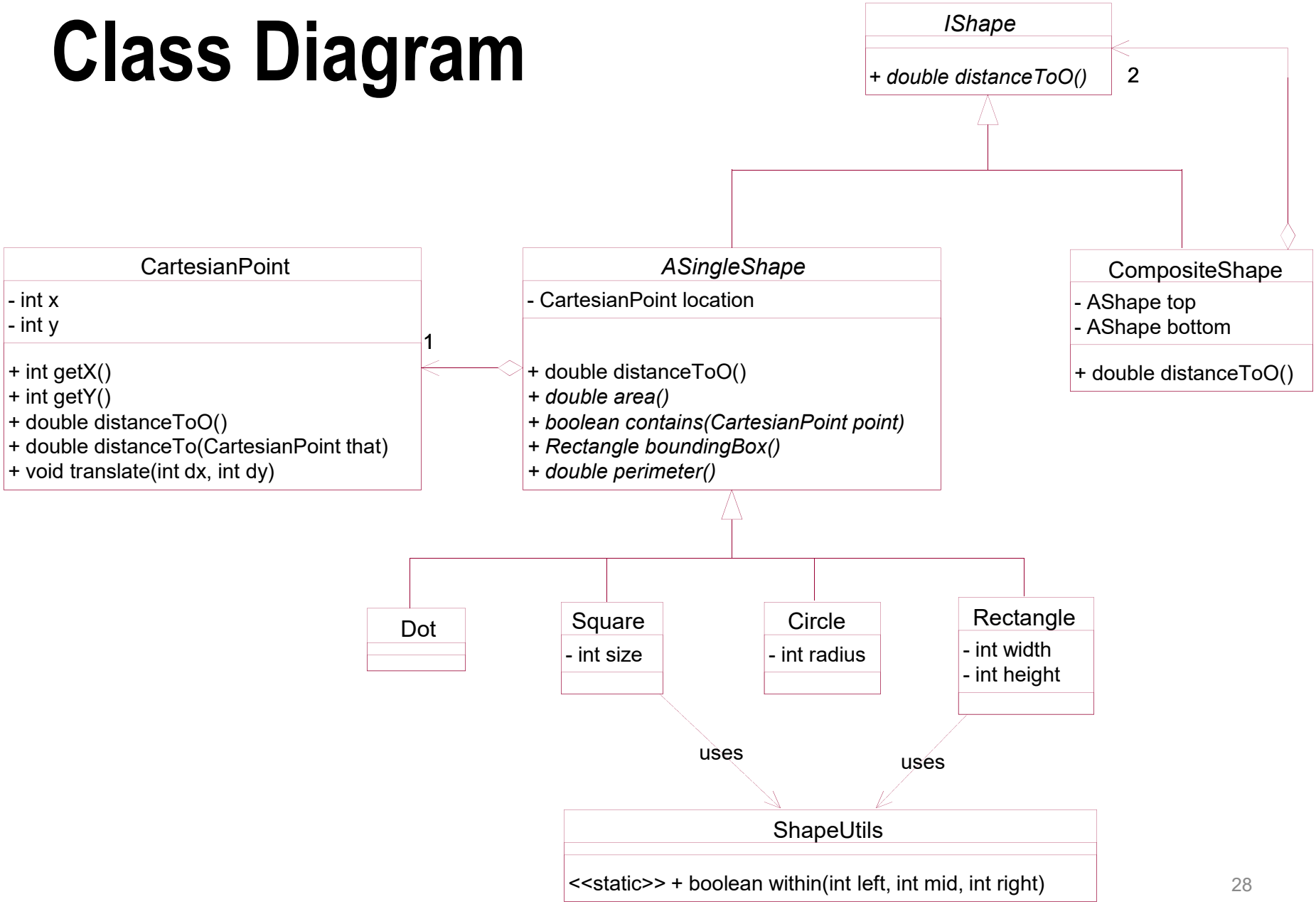
Excercise 5.10: Overlapping Shapes

- Develop a drawing program that deals with at least three kinds of shapes: dots, squares, and circles. ...In addition, the program should also deal with overlaying shapes on each other. In the following figure, for example, we have superimposed a circle on the right side of a square:



- We could now also superimpose(thêm vào) this compounded shape on another shape and so on.

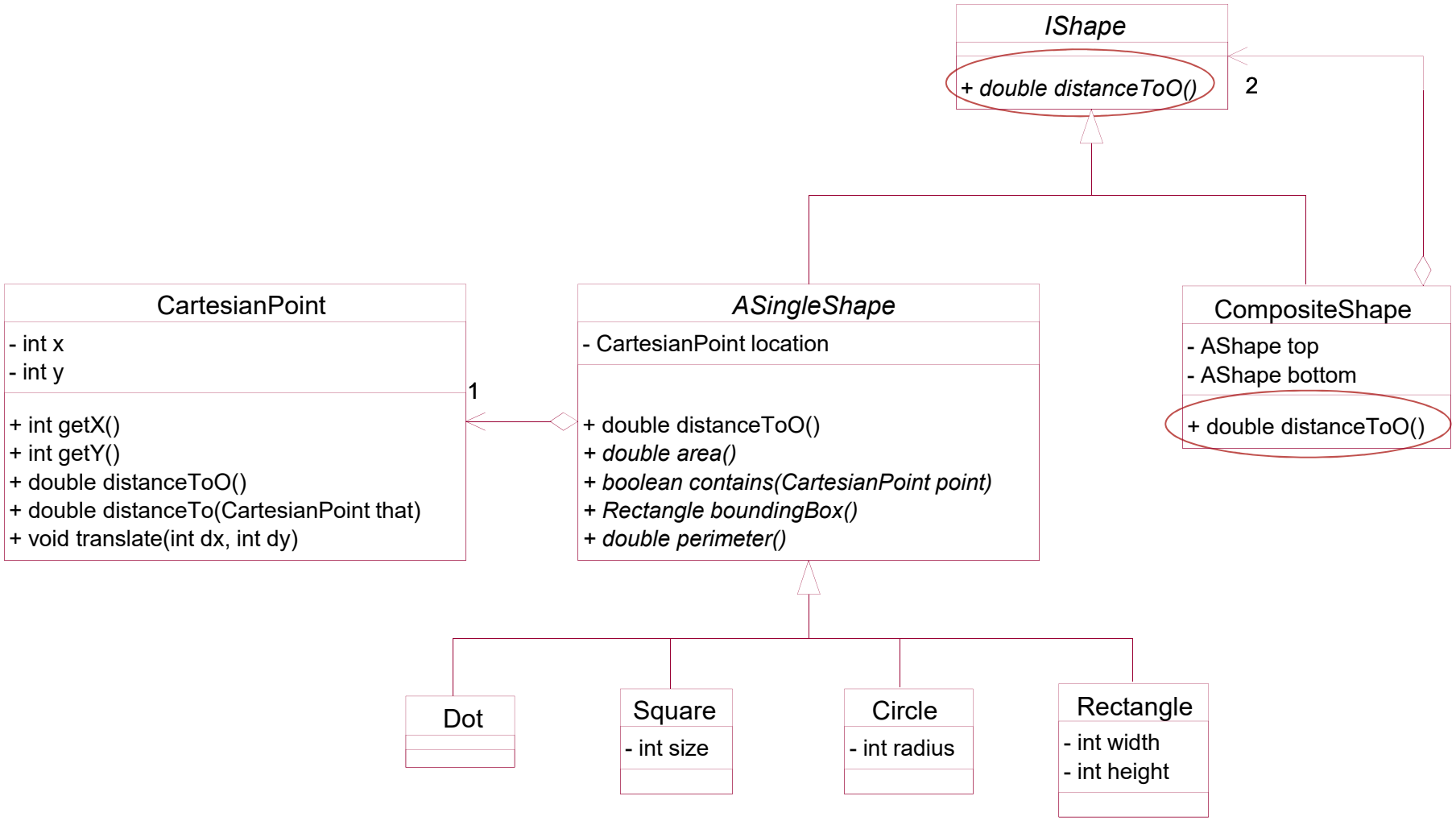
Class Diagram



Problem1

- ... The user wishes to know how close a combination of shapes is to the origin ...

Modified Class Diagram



distanceToO() in CompositeShape

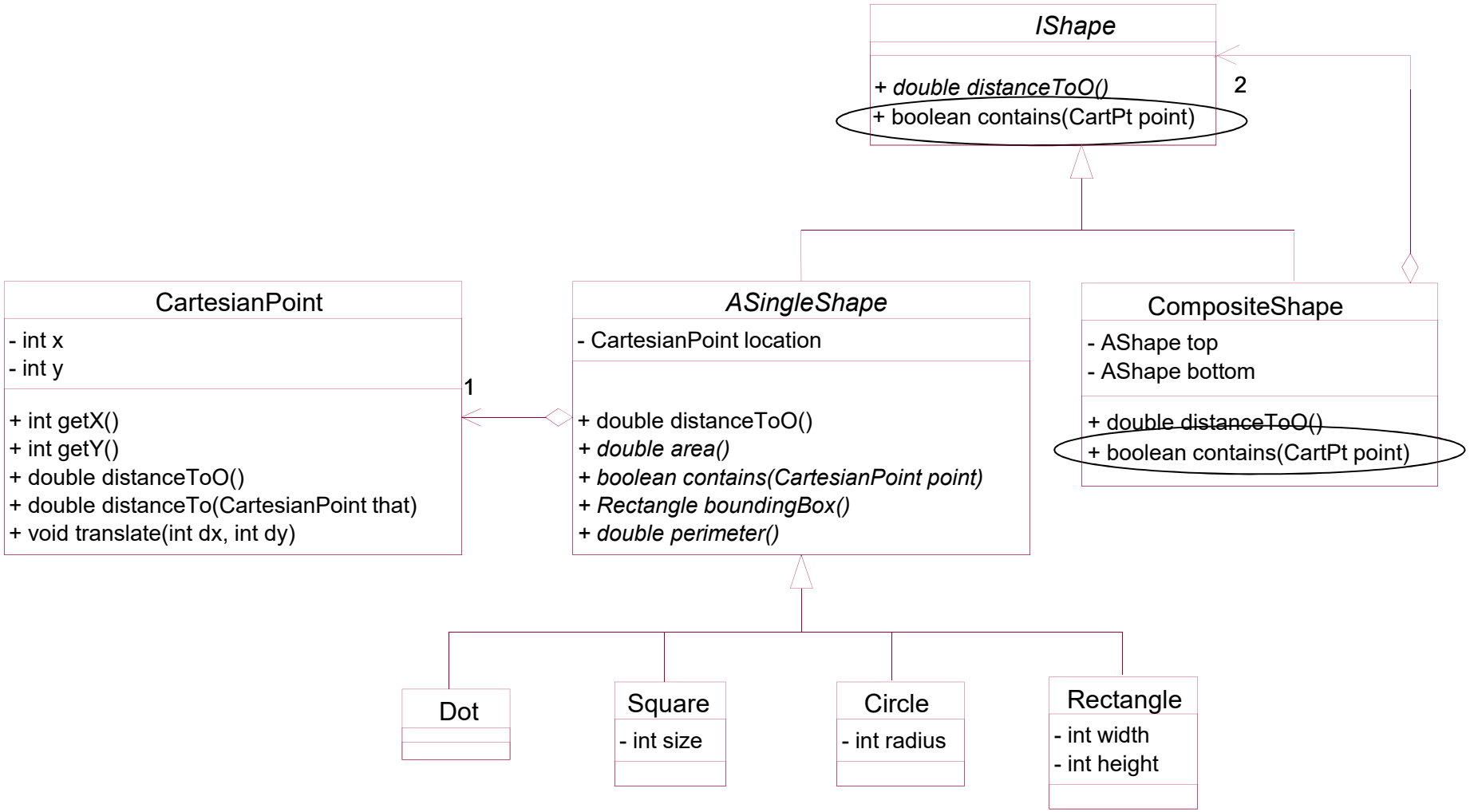
```
public class CompositeShape implements IShape {
   private IShape top;
   private IShape bottom;
   public CompositeShape(IShape top, IShape bottom) {
      this.top = top;
      this.bottom = bottom;
   }

   public double distanceToO() {
      return Math.min(this.top.distanceToO(),
                      this.bottom.distanceToO());
   }
}
```

Problem 2

- ... Add a method that determines whether some point in the Cartesian space falls within the boundaries of some shape. ...

Modified Class Diagram



contains() in CompositeShape

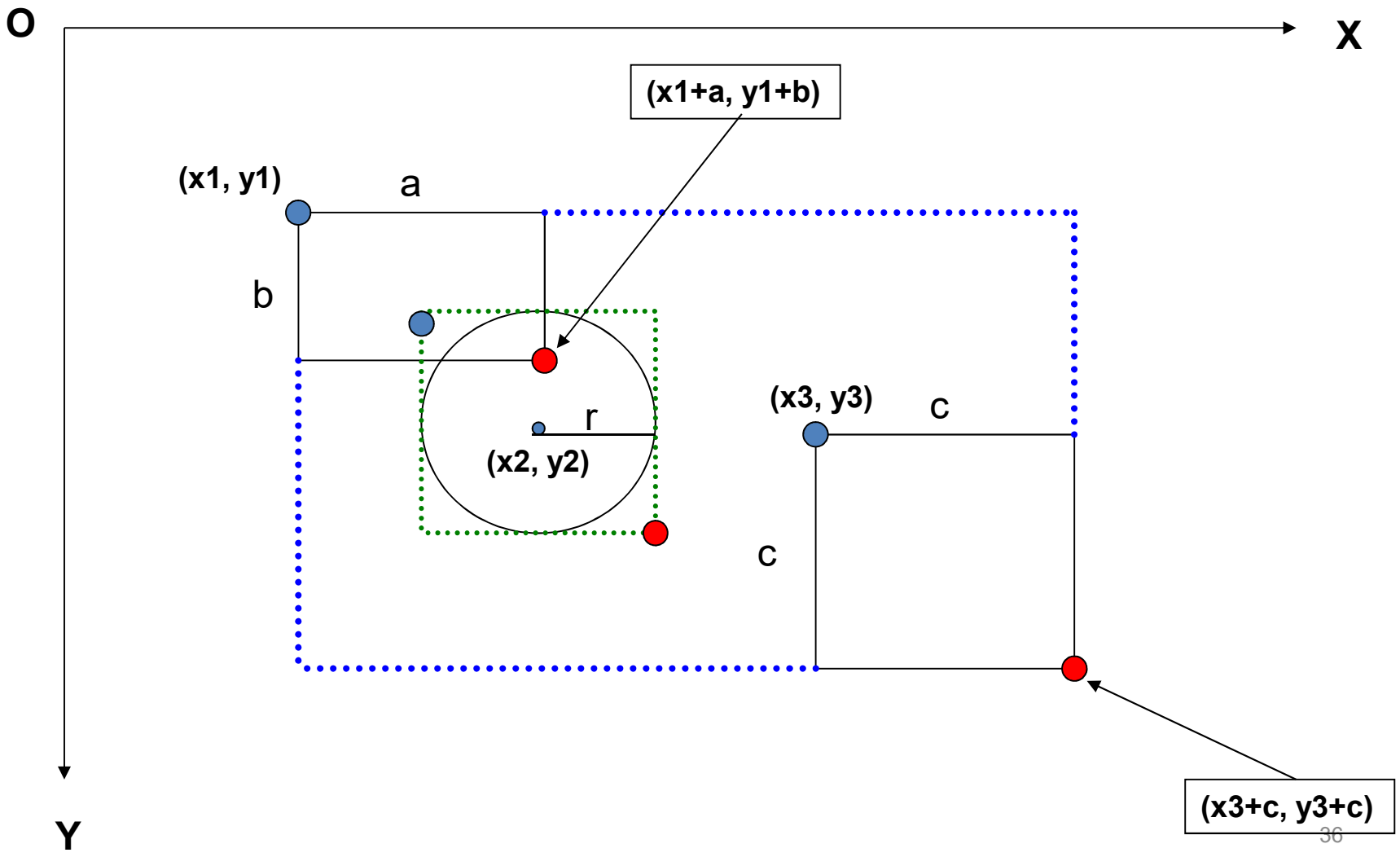
```
public class CompositeShape implements IShape {
   private IShape top;
   private IShape bottom;
   public CompositeShape(IShape top, IShape bottom) {
      this.top = top;
      this.bottom = bottom;
   }

   public boolean contains(CartPt point) {
      return this.top.contains(point)
             || this.bottom.contains(point);
   }
}
```

Problem 3

- ... A graphics program must compute the bounding box for a shape. ...

Bounding box for Composite shape



boundingBox() Examples

```
IShape s1 = new Square(new CartPt(4, 3), 40);
IShape s2 = new Square(new CartPt(3, 4), 50);
IShape c1 = new Circle(new CartPt(0, 0), 20);
IShape c2 = new Circle(new CartPt(12, 5), 20);
IShape u1 = new CompositeShape(s1, s2);
IShape u2 = new CompositeShape(s1, c2);
IShape u3 = new CompositeShape(c1, u1);
IShape u4 = new CompositeShape(u3, u2);

s1.boundingBox() should be new Rectangle(new CartPt(4, 3), 40, 40)
s2.boundingBox() should be new Rectangle(new CartPt(3, 4), 50, 50)
c1.boundingBox() should be new Rectangle(new CartPt(-20, -20), 40, 40)
c2.boundingBox() should be new Rectangle(new CartPt(-8, -15), 40, 40)
u1.boundingBox() should be new Rectangle(new CartPt(3, 3), 50, 51)
u2.boundingBox() should be new Rectangle(new CartPt(-8, -15), 52, 58)
u3.boundingBox() should be new Rectangle(new CartPt(-20, -2), 73, 74)
u4.boundingBox() should be new Rectangle(new CartPt(-20, -20, 73, 74)
```

boundingBox() in CompositeShape

```
public Rectangle boundingBox() {
   Rectangle bbTop = this.top.boundingBox();
   Rectangle bbBottom = this.bottom.boundingBox();
   int x1 = Math.min(bbTop.location.getX(),
                     bbBottom.location.getX());
   int y1 = Math.min(bbTop.location.getY(),
                     bbBottom.location.getY());
   int x2 = Math.max(bbTop.location.getX() + bbTop.getWidth(),
               bbBottom.location.getX() + bbBottom.getWidth());
   int y2 = Math.max(bbTop.location.getY() + bbTop.getHeight(),
               bbBottom.location.getY() + bbBottom.getHeight());
   return new Rectangle(new CartPt(x1, y1),
                x2 - x1, y2 - y1);
}
```

Exercise 5.11

Develop a program that sorts lists of mail messages by date.
Mail structures are defined as follows: from, date, message

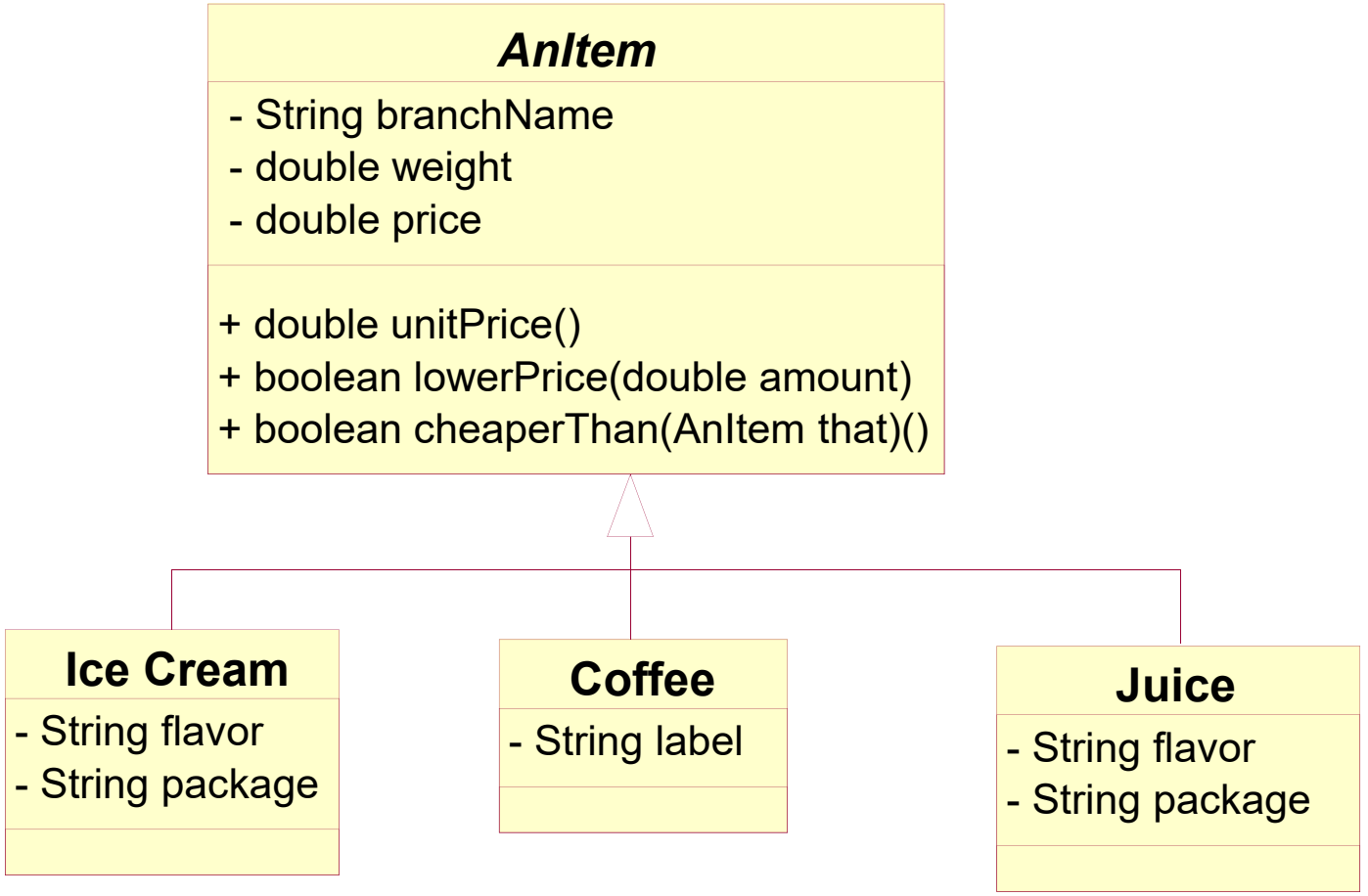
Exercise 5.12

Design a data representation for shopping lists.

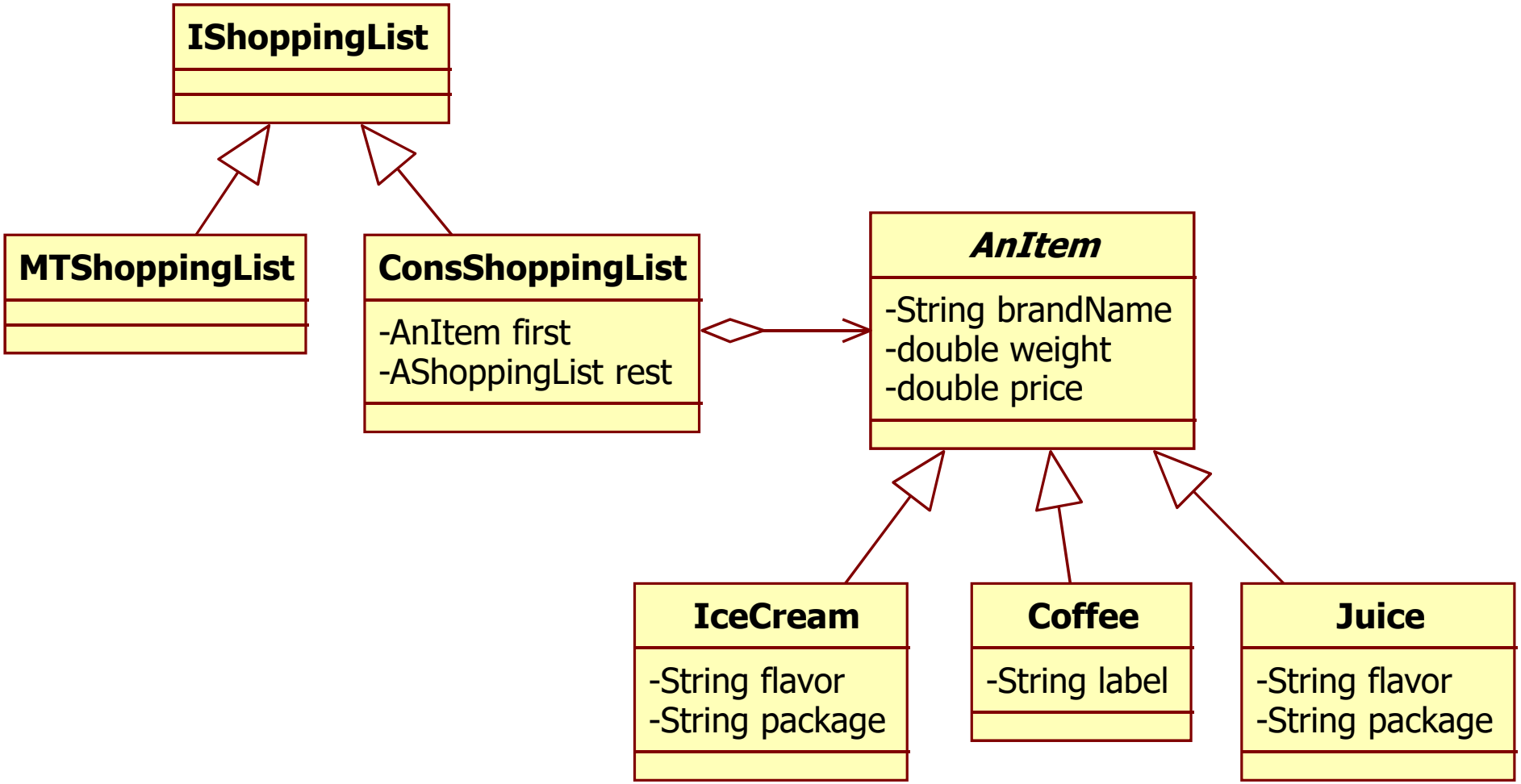
Start from the class of grocery items developed in exercise 4.6. Add the following methods:

- howMany, which computes the number of items on the shopping list;
- brandList, which produces the list of all brand names;
- highestPrice, which determines the highest unit price among all items in the shopping list.

Exercise 5.12 Class Diagram



ShopingList class diagram

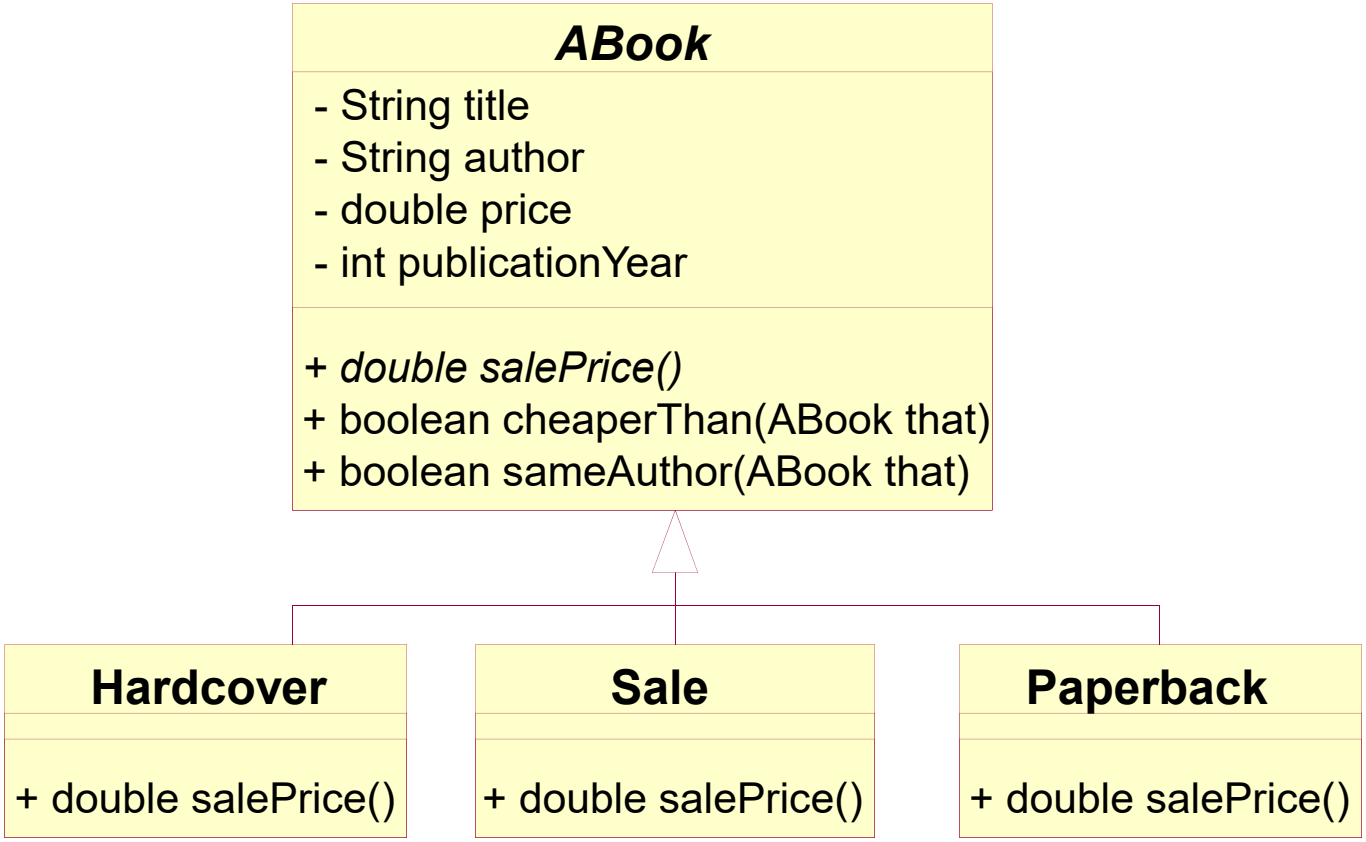


Exercise 5.13

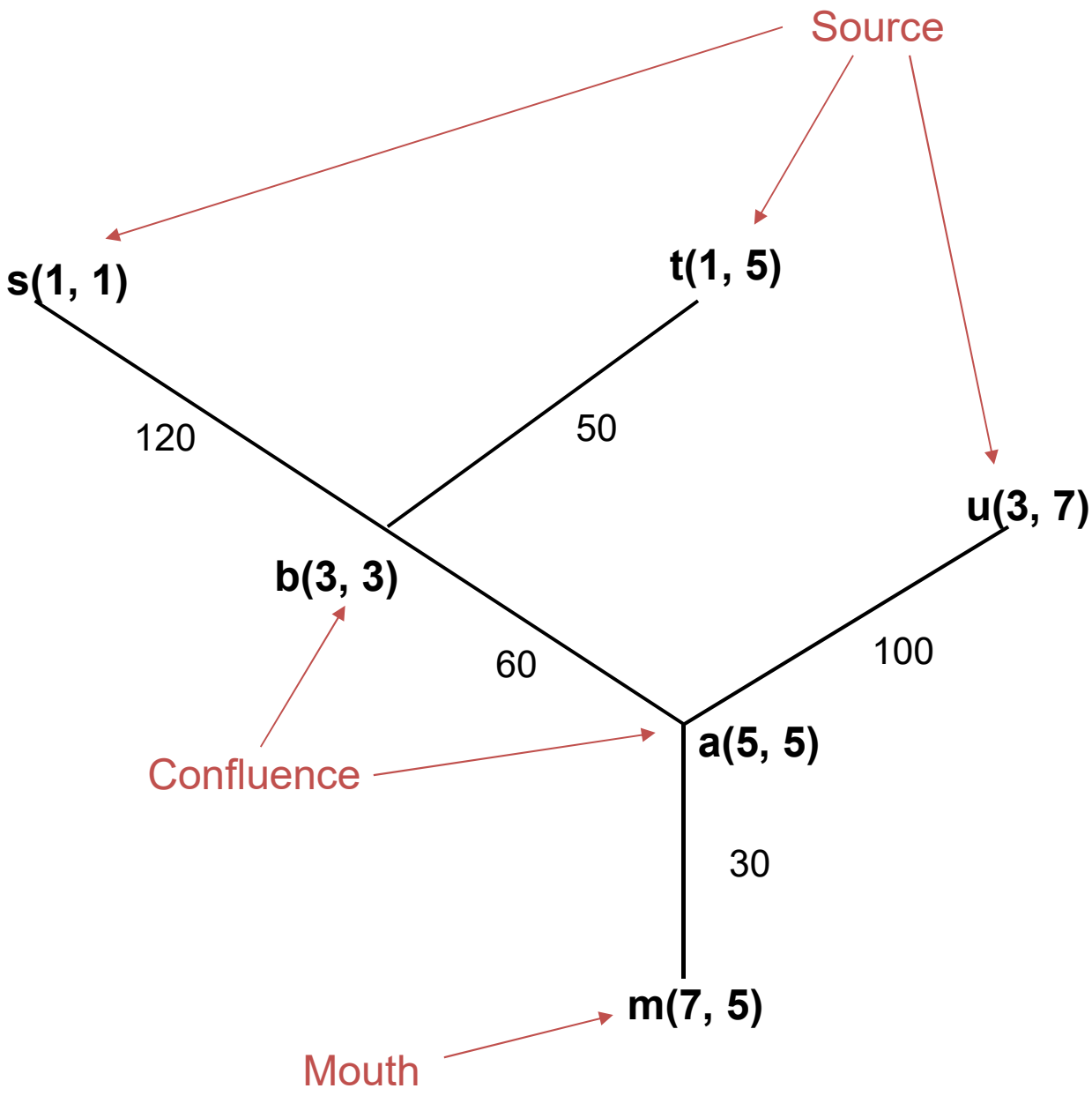
Develop a program for managing discount bookstores (see exercise 5.2):

- Design a representation for lists of books;
- Write down (in English) three examples of book lists and their corresponding data representations;
- Develop the method thisAuthor, which produces the list of books that this author has authored.
- Develop the method sortByTitle, which sorts lists of books by title

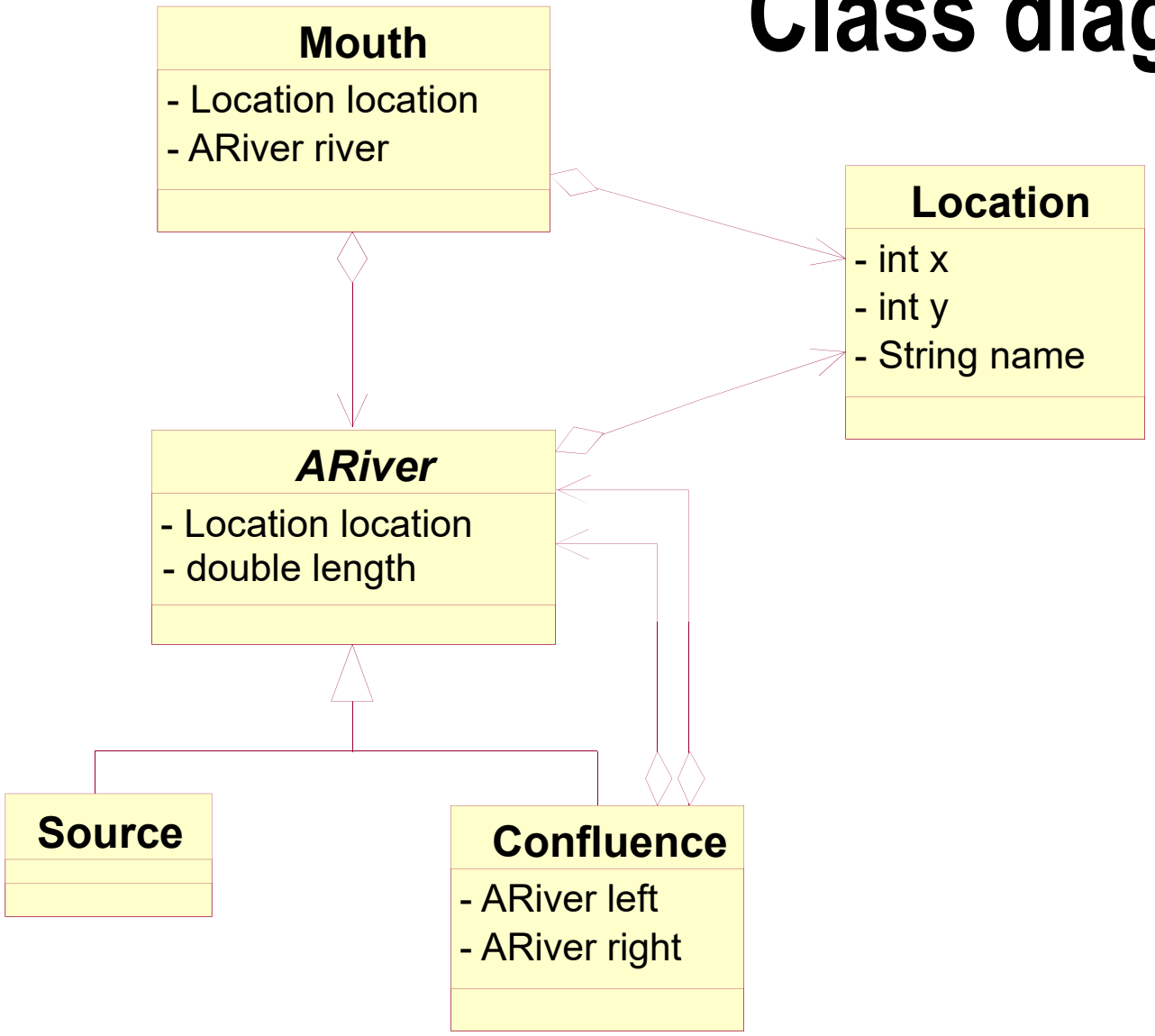
Exercise Class Diagram



Exercise 5.14: River Systems Example


Source
s(1, 1)
t(1, 5)
120
50
u(3, 7)
b(3, 3)
60
100
Confluence
a(5, 5)
30
m(7, 5)
Mouth

Class diagram



Problems

- The EPA must represent river systems and monitor them… An EPA officer may wish to query a computer about the number of sources that feed a river system.
- An EPA officer may wish to find out whether some location is a part of a river system, regardless of whether it is a source, a confluence, or the river mouth.
- An EPA officer may request the number of miles of a river system, either starting from the river's mouth or any of its confluence points.
- An EPA officer may wish to find out whether some location is within a given radius of some confluence or source on a river system

Problems

Extend the following methods to classes that represent river systems with the following methods:

- `maxlength`, which computes the length of the longest river segment;
- `confluences`, which counts the number of confluences in the river system; and
- `locations`, which produces a list of all locations on this river -- the sources, the mouths, and the confluences.