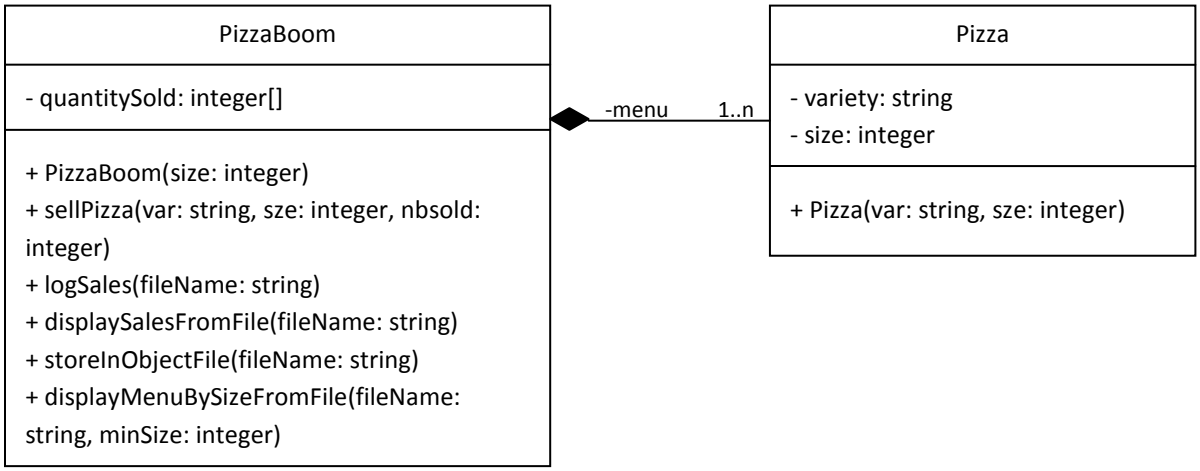
Name:	Section:	Spring 2010
ID:	CSC113 MidTerm-2	Dr.

EXERCISE 1

Translate into java the following UML class diagram. The *PizzaBoom* class represents a pizzeria and the *Pizza* class represents a type of pizza that belongs to the pizzeria's menu.



Pizza class attributes

- ***variety:*** the pizza variety. For example: mexican, vegetarian, etc.
- ***size***: means the pizza size. For example: 6 inch, 12 inch, etc.

***PizzaBoom* class attributes**

- ***menu***: array that contains the pizzeria's menu
- ***quantitySold***: array that contains the number of sold pizza for each pizza type. The index in *quantitySold* corresponds to the index of pizza type in *menu*.

Example: the quantity of pizza sold is: 345 of mexican 12 inch, 187 of mexican 6 inch, and 59 of vegetarian 5 inch.

menu

mexican 12	mexican 6	vegetarian 5

quantitySold

345	187	59

***PizzaBoom* class methods**

- ***sellPizza***: when pizza is sold, this method receives the following parameters:
  - ***var***: the pizza variety
  - ***sze***: the pizza size
  - ***nbSold***: the number of pizza sold

  The method updates the array *quantitySold* by adding *nbSold.*

  Example: we sold 4 of vegetarian 5 inch pizza. *sellPizza* will add 4 to 59 to become 63.

- ***logSales***: writes the content of the arrays *menu* and *quantitySold* in a text file using *PrintWriter* and the format of the following example:

Pizza type	Size	Quantity
mexican	12	345
mexican	6	187
vegetarian	5	63

- ***displaySalesFromFile****:* reads the text file created by *logSales* and displays that content on the screen.
- ***storeInObjectFile****:* writes the content of the *menu* array in a file of objects.
- ***displayMenuBySizeFromFile***: reads the file of objects created by *storeInObjectFile* and displays on the screen the pizza types whose size is greater than ***minSize***.
  Example, if *minSize* equals 5, this method should display on the screen:
  mexican 12
  mexican 6

Answer

Name:	Section:	Spring 2010
ID:	CSC113 MidTerm-2	Dr.

EXERCISE 2

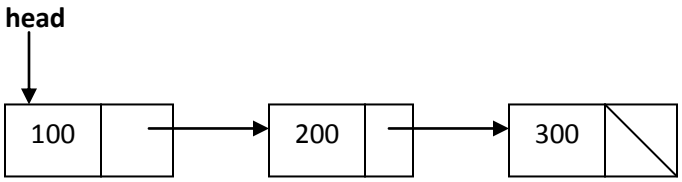
The java code method below has been executed with the following series of input data (through the scanner): t444, 4t44, 44t4, 4444t, 4444. Write the content of the output console.

```
public void inputSalary()
{
      String stsal = "0";
      int salary = 0;
      boolean stop = false;
      Scanner input = new Scanner(System.in);
      System.out.println("Enter the salary\n");
      stsal = input.next();
      while(!stop)
      {
            try
            {
                  salary = Integer.parseInt(stsal);
                  stop = true;
            }
            catch(NumberFormatException e)
            {
                  System.out.println("Invalid salary " + stsal);
                  stsal = input.next();
            }
      }
      System.out.println("Salary is: " + salary);;
}
```

Name:	Section:	Spring 2010
ID:	CSC113 MidTerm-2	Dr.

EXERCISE 3

A/ Assume you have the following list:

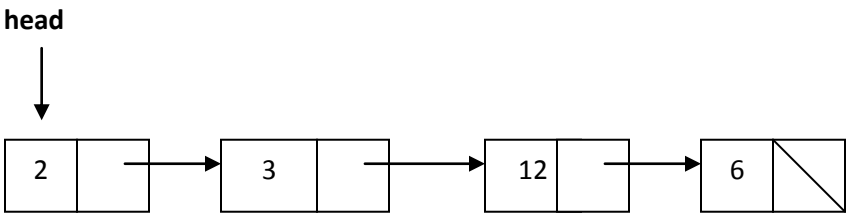


Draw a picture of the list after the following code is executed:

```
Node curr = head;
while (curr.getNext()!= null)
{
    curr = curr.getNext();
}
curr.setNext(head);
head = head.getNext();
curr = curr.getNext();
curr.setNext(null);
```

B/ Write a method *boolean InsertAfterElement(int value1, int value2)* method which inserts *value1* after the first element greater than *value2*. Consider special cases: 1) list empty 2) no element in the list is greater than *value2*.

Example:



InsertAfterElement(77, 10) means to insert 77 after the first element greater than 10: