```
import java.awt.Point;
import java.io.File;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.Scanner;

/**
 * This is an helper class that is used to perform different operations
 * using a list of points.
 */
public class PointProcessor {

    public static ArrayList<Point> readPointsFromFile(String fileName) {

      ArrayList<String> lines = new ArrayList<String>();
      Scanner file = null; // isnt initalized because it needs to be in try catch

      try { //tries to scan the input file and add strings to the arraylist of 
lines
            file = new Scanner(new File(fileName));

            while(file.hasNextLine()) {
                  lines.add(file.nextLine());
            }

      }catch(FileNotFoundException e){ // if file stated doesnt exist it catches 
the error and reports it
            e.printStackTrace();
      }catch(Exception e) { // if there are any other errors for any reason it will 
report it
            e.printStackTrace();
      }finally {
            try {
                  file.close(); // closes the scanner to prevent data loss
            }catch(Exception f){ // if there is an error closing the scanner it 
will report it
                  f.printStackTrace();
            }
      }


      ArrayList<Point> points = new ArrayList<Point>(); // creates an arraylist for 
points to be put in

      for(String s : lines) { //loops through each instance of a string in the 
arraylist lines
            String[] parts = s.split(" "); // splits the string in the arraylist 
where a space occurs
            int x = Integer.parseInt(parts[0]); //turns string into an int x
            int y = Integer.parseInt(parts[1]); // turns string into an int y
            Point p = new Point(x,y) ; //makes a new point with freshly made 
integers x,y
            points.add(p); //adds the point to the points arraylist
      }

        return points;   // returns array of points
    }

    public static int cabDistance(Point pt) {
      double pointx = pt.getX(); //gets point x as a double
            double pointy = pt.getY();// gets point y as a double
            int total = (int) Math.abs((int)pointx) + (int) Math.abs((int)pointy) ;
// calculates their total
            int distance = Math.abs(0-total); // calculates distance from (0,0)

      return distance;  // returns distance from 0
    }


    public static void showPoint(Point pt) {
      double pointx = pt.getX(); // gets x value of the point for use in printing
      double pointy = pt.getY(); // gets y value of the point for use in printing
      int x = (int) pointx; // casts the double pointx as an int for printing use
      int y = (int) pointy; // casts the double pointy as an int for pringing use
      System.out.printf("(" + x + "," + y + ")" + "\t" + cabDistance(pt)); //
prints out the desired information

      return;
    }

    public static void showAllPoints(ArrayList<Point> ptList) {
      if(ptList.size() == 0) { // checks to see if list is empty
            System.out.println("The list is empty");//if it is it will alert the
user that it is empty
      }else { // it it is not empty it will proceed as follows
            for(int i = 0; i<ptList.size(); i++) { // loops through list

                  System.out.printf("\n" + "[" + i + "]" + " "); //prints out index
of the point
                  showPoint(ptList.get(i)); // prints out the points information
            }
      }
    }

    public static ArrayList<Point> findAll(ArrayList<Point> ptList, int dist) {

      ArrayList<Point> distList = new ArrayList<Point>(); // makes an array to
return
      for(int i = 0; i<ptList.size(); i++) { // loops through ptList
            Point pt = ptList.get(i); // gets the point from the ptList
            int distance = cabDistance(ptList.get(i));// gets the distance of
certain point by use of cabDistance method
            if(distance == dist) { // checks if distance from 0 is what we want
            distList.add(pt); // adds the point to the array if it is the distance
we want
            }
      }
        return distList;   // returns the completed array
    }
}
```