```
/*
 * CS5405
 * Homework 05
 * Grant Broadwater (grbcp5)
 * October 16, 2018
 *
 * CircleDataModel.java
 */

package code;

import java.util.Scanner;
import java.io.File;
import java.io.FileNotFoundException;
import java.util.LinkedList;

public class CircleDataModel {

  private DemoCircle circle1;
  private DemoCircle circle2;

  private DemoCircle[][] testCases;

  public CircleDataModel() {
    this.circle1 = new DemoCircle(50, 50, 10);
    this.circle2 = new DemoCircle(100, 50, 10);

    this.testCases = this.getTestCasesFromFile();
  }

  private DemoCircle[][] getTestCasesFromFile() {
    LinkedList<DemoCircle[]> result = new LinkedList<DemoCircle[]>();;
    Scanner fileScanner;
    Scanner strScanner;

    try {
      fileScanner = new Scanner(new File("data.txt"));
    } catch(FileNotFoundException e) {
      try {
        fileScanner = new Scanner(new File("data/data.txt"));
      } catch(FileNotFoundException f) {
        try {
          fileScanner = new Scanner(new File("data/inFile.txt"));
        } catch(FileNotFoundException g) {
          return new DemoCircle[0][0];
        }
      }
    }


    while(fileScanner.hasNextLine()) {
      DemoCircle[] newTestCase = new DemoCircle[2];
      strScanner = new Scanner(fileScanner.nextLine());
      newTestCase[0] = readCircleFromScanner(strScanner);
      newTestCase[1] = readCircleFromScanner(strScanner);
      result.addLast(newTestCase);
    }

    return result.toArray(new DemoCircle[result.size()][2]);
  }


  private DemoCircle readCircleFromScanner(Scanner circleScanner) {
    return new DemoCircle(
      Double.parseDouble(circleScanner.next().replace(",", "")),
      Double.parseDouble(circleScanner.next().replace(",", "")),
      Double.parseDouble(circleScanner.next().replace(",", ""))
    );
  }


  public void setCircleData(String circleData) {
    Scanner circleScanner = new Scanner(circleData);

    this.circle1 = readCircleFromScanner(circleScanner);
    this.circle2 = readCircleFromScanner(circleScanner);
  }


  public void setCircleData(DemoCircle[] circleData) {
    this.circle1 = circleData[0];
    this.circle2 = circleData[1];
  }


  public int getNumTestCases() {
    return this.testCases.length;
  }


  public DemoCircle[] getTestCase(int testCaseIdx) {
    if (0 <= testCaseIdx && testCaseIdx < this.getNumTestCases()) {
      return this.testCases[testCaseIdx];
    }
    return null;
  }


  public javafx.scene.shape.Circle getCircle1() {
    return this.circle1;
  }
  public javafx.scene.shape.Circle getCircle2() {
    return this.circle2;
  }


  public String getCircleRelationship() {

    if (this.circle1.equals(circle2)) {
      return "Circles are identical.";
    } else if (this.circle1.isInside(this.circle2)) {
      if(this.circle1.isInternallyTouching(this.circle2)) {
        return "C1 is inside of and touching C2.";
      } else {
        return "C1 is inside of but not touching C2.";
      }
    } else if (this.circle2.isInside(this.circle1)) {
      if(this.circle2.isInternallyTouching(this.circle1)) {
        return "C2 is inside of and touching C1.";
      } else {
        return "C2 is inside of but not touching C1.";
      }
    } else if (this.circle1.isOutside(this.circle2)) {
      if(this.circle1.isExternallyTouching(this.circle2)) {
        return "C1 is outside of and touching C2.";
      } else {
        return "C1 is outside of but not touching C2.";
      }
    }else {
      return "Circles properly overlap.";
    }
  }


  public String toString() {
    return this.circle1.getCenterX()
            + " " + this.circle1.getCenterY()
            + " " + this.circle1.getRadius()
            + " " + this.circle2.getCenterX()
            + " " + this.circle2.getCenterY()
            + " " + this.circle2.getRadius();
  }
}
```